

68

\$2.95_{USA}

Australia
Singapore
Malaysia

A \$ 4.75
S \$ 9.45
M \$ 9.45

New Zealand NZ \$ 6.50
Hong Kong H \$ 23.50
Sweden 30.-SEK

MICRO JOURNAL

VOLUME V ISSUE XI • Devoted to the 68XX User • November 1983
"Small Computers Doing Big Things"

SERVING THE 68XX USER WORLDWIDE



PHOTO CREDIT: NASA





FREE FROM LIMITATIONS, A TEACHER CAN UNLOCK IMAGINATIONS.

By 1985, fully 75% of all jobs will involve computers. So for today's teachers, selecting the right computer is an awesome responsibility. It's a decision you can feel confident about with Southwest Technical Products Corporation's S+ computer system. The S+ is specially designed for a student's tomorrow and a teacher's today. It's built tough. It's simple to operate. You can teach without becoming a hardware specialist.

The S+ is fluent in a variety of languages: BASIC, PASCAL, COBOL, FORTRAN, PILOT,

FORTH, "C" and MUMPS and allows you simultaneous use in the classroom. This flexibility enables you to use your own curriculum, not a machine's version. The system also allows for individual learning styles. Students go as fast and as far as the teacher chooses.

Even though it's priced like a smaller personal computer, the S+ has the multi-task power and expansion abilities of a professional computer. It can grow with you and your students.

Tomorrow is in your hands,

today. Call Fisher Scientific, exclusive distributor of our S+ system, and you'll get the key to unlock imaginations.

For more information. Call toll-free 800-621-4769. Illinois customers call collect (312) 378-7770.

Fisher Scientific

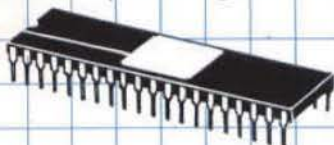
An  **ALLIED** Company

Educational Materials Division
4901 W. LeMoyné St.,
Chicago, Illinois 60651

MICROWARE'S OS-9 IS NUMBER ONE.

OS-9 NOW HAS THE LARGEST USER COMMUNITY

More users now run OS-9 on their 6809 computers than all other operating systems combined. This outstanding success story was no accident — it's due to OS-9's technical excellence backed up by outstanding Microware support. OS-9's Unix-type architecture and totally modular design gives your computer more power and versatility. OS-9 also gives you more possibilities for customization so you can tailor your system exactly to your needs. And aren't flexibility and performance the reasons you chose a 6809 computer to begin with?



OS-9 HAS BEEN CHOSEN BY OVER 50 6809 SYSTEM MANUFACTURERS

OS-9 is now offered as a standard operating system by almost every 6809 system manufacturer, and has been designed into an amazing variety of dedicated systems and products including personal and business computers, process control systems, data and telecommunications systems, and more. In all, over 50 companies and organizations have

obtained OS-9 distribution licenses including such well-known names such as General Motors, NASA, Fujitsu, Western Electric, Motorola, Sykes Datatronics, Eastman Kodak, Thomson-CSF, and Tandy Corp.

OS-9 GIVES YOU A SOFTWARE BASE TO BUILD ON

Whatever your application, OS-9 speaks your language! Microware offers BASIC09, an Extended/Structured Basic, a complete C Compiler, a full ISO Pascal Compiler, the ANSI Standard CIS Cobol Compiler, plus Relocatable Macro Assembler. These high performance programming languages are all fully implemented and deliver unmatched performance and outstanding features. Additionally, OS-9 compatible applications packages such as word processors, screen editors, spreadsheets, business software, and utilities are offered by a rapidly growing number of third-party software houses.

PLUS OUTSTANDING MICROWARE SUPPORT: WE KEEP IN TOUCH WITH YOU

Even when you have the best software and documentation, there can be times when you need questions answered. That's why Microware is committed to giving OS-9 users the best possible personalized service. Here are some

of the ways we deliver solid support:

- A Software Support Hotline for direct access to our technical staff
- "Pipelines", our free quarterly newsletter
- OS-9 User Seminars, the annual OS-9 community gathering
- a liberal update policy for new releases

Microware does business on a person-to-person basis. When you call you'll find yourself speaking with someone who's both knowledgeable and genuinely interested in helping.

YOU CAN COUNT ON OS-9 NOW AND IN THE FUTURE

Microware is not standing still — we're firmly committed to continuing support for the 6809 and we will continue to introduce exciting new software products for the 6809. We will soon announce OS-9/68000 and programming languages for the 68000 which will be upward compatible with 6809 versions, so if and when you are ready for the 68000 your OS-9 software can go with you.



MICROWARE

Microware Systems Corporation
Box 4865 • Des Moines, IA 50304
515-279-8844 Telex 910-520-2535

OS-9 and Basic09 are trademarks of Microware and Motorola.
This is a trademark of Bell Labs.

OS-9: BETTER BY DESIGN

'68'

MICRO JOURNAL

Portions of the text for 68 MICRO JOURNAL was prepared using the following furnished hard/software.

COMPUTERS-HARDWARE

Southwest Technical Products
219 W. Rhapsody
San Antonio, Texas 78216
S09-5/8 DMF disk-CDS1-8212W-Sprint 3 Printer

GIMIX Inc.
1337 West 37th Place
Chicago, IL 60609
Super Mainframe-OS9-FLEX-Assorted Hardware

EDITORS-WORD PROCESSORS

Technical Systems Consultants, Inc.
111 Providence Road
Chapel Hill, NC 27514
FLEX-Editor-Processor

Great Plains Computer Company, Inc.
PO Box 916
Idaho Falls, ID 83401
STYLO-Mail Merge

Editorial Staff

Don Williams Sr.	Publisher
Larry E. Williams	Executive Editor
Tom E. Williams	Production Editor
Robert (Bob) Nay	Color Editor

Administrative Staff

Mary Robertson	Office Manager
Joyce Williams	Accounting
Carolyn Williams	Subscriptions
Penny Williams	File Management

Contributing Editors

Ron Anderson
Norm Cammo
Peter Dibble
Dr. Theo Elbert
William E. Fisher
Or. E.M. Pass

Special Technical Projects

Clay Abrams K6AEP
Tom Hunt

CONTENTS

Vol. V, Issue XI

NOVEMBER '83

FLEX USER NOTES.....	10	Anderson
OS9 USER NOTES.....	11	Dibble
BT-1 GRAPHICS BOARD REVIEW.....	16	Hanon
NEW COPY UTILITY.....	18	Taylor
		& Puglia
META LAB DAC1220 REVIEW.....	33	Ferguson
TSC ASSEMBLER PRINT UTILITY....	34	Cole
XDMS DATA MANAGEMENT SYSTEM....	35	Staff
BIT BUCKET.....	39	
UPGRADING A PERCOM SYSTEM.....	39	Bollinger
NOT JUST ANOTHER DUMB ROUTINE..	41	Rees-Thomas
"HOLD" REVISED.....	43	Korte
A SMALL C.....	45	STS
CLASSIFIED ADVERTISING.....	49,56	
ADVERTISERS INDEX.....	70	

Send All Correspondence To:

Computer Publishing Center
68 MICRO JOURNAL
5900 Cassandra Smith
PO Box 849
Hixson, TN 37343
615 842-4600

Copyrighted 1983 by Computer Publishing, Inc. (CPI)

68' Micro Journal is published 12 times a year by Computer Publishing Inc. Second Class Postage Paid ISSN 0194-5025 at Hixson, Tenn. and additional entries. Postmaster: send Form 3579 to 68' Micro Journal, PO Box 849, Hixson, Tennessee.

SUBSCRIPTION RATES

USA

1-Year \$24.50 2-Years \$42.50 3-Years \$64.50

FOREIGN

See Page 52

Items Submitted for Publication

Articles submitted for publication should be accompanied by the authors full name, address, date and telephone number. It is preferred that articles be submitted on either 5 or 8 inch diskette in TSC Editor format or STYLO format. All diskettes will be returned.

The following TSC Text Processor commands ONLY should be used (due to our proportional processor): .sp space, .pp paragraph, .fl fill and .nf no fill. Also please do not format within the text with multiple spaces. The rest we will enter at time of editing.

STYLO commands are all acceptable except the .pg page command, we print edited text files in continuous text.

All articles submitted on diskettes should be in TSC FLEX" format, either FLEX2 6800, or FLEX9 6809 any version.

If articles are submitted on paper they should be on white 8X11 bond or better grade paper. No hand written articles (hand written or drawn art accepted). All paper submitted articles will be photo reproduced. This requires that they be typed or produced with a dark ribbon (no blue), single spaced and type font no smaller than 'elite' or 12 pitch. Typed text should be approximately 7 inches wide (will be reduced to column width of 3 1/2 inches). Please use a dark ribbon!

All letters to the editor should also comply with the above and bear a signature. Letters of 'gripes' as well as 'praise' are solicited. We attempt to publish all letters to the editor verbatim, however, we reserve the right to reject any submission for lack of 'good taste'. We reserve the right to define what constitutes 'good taste'.

Advertising: Commercial advertisers please contact the 68 Micro Journal advertising department for current rate sheet and requirements.

Classified: All classified must be non-commercial. Maximum 20 words per classified ad. Those consisting of more than 20 words should be figured at .35 cents per word, 20 words or less \$.75 minimum, one time, paid in advance. No classified ads accepted by telephone.

TO REALIZE THE FULL POWER & PERFORMANCE OF THE 6809, LOOK TO GIMIX.

GIMIX OFFERS YOU A VARIETY OF SS50 BUS COMPONENTS AND SYSTEMS.

OS-9 GMX III

The **GMX 6809 CPU III** and **OS-9 GMX III**. A Multi-user, Multi-tasking package for the ultimate in System Performance plus protection of the system and other users from crashes caused by errors in individual users programs.

NEW! \$1088.01
INTELLIGENT I/O PROCESSOR BOARDS increase system throughput by reducing interrupts to the host, buffering data transfers, and data preprocessing. Prices include on board firmware. Requires system drivers.
 #11 3 port RS232 Serial (SS30) **NEW!** \$498.11
 #12 4 port Parallel (SS50) \$538.12
 OS-9 GMXIII drivers. (Included when purchased with GMX III package). \$200.00
 OS-9 Level 2 users -- contact GIMIX for system requirements and availability.

192K GMX III #79 SYSTEMS: All include GMX 6809 CPU III and OS-9 GMX III (#01): a #11 3 port intelligent serial I/O & cables; #19 Classy Chassis; 192KB Static RAM, #68 DMA controller, all necessary cables, power regulators, and filter plates. The OS-9 Editor, Assembler, Debugger, BASIC-09, and RUNB are included.
 #79 with dual 40 track DSDD drives **NEW!** \$5508.79
 #79 with dual 80 track DSDD drives \$6298.79
 #79 with #88 8" Dual Drive Disk System \$7598.79
 #79 with #90 19MB Winchester subsystem & one 80 track DSDD drive \$8898.79
 UnIFLEX for the GMX 6809 CPU III and intelligent I/O boards is in development.

OS-9 GMX I; OS-9 GMX II; FLEX; and UnIFLEX

The #05 GIMIX 6809 PLUS CPU board \$578.05
 Options: GMX DAT \$35.00 SWTPC DAT \$15.00
 9511A \$312.00 9512 \$265.00

#49 64KB GHOST SYSTEM includes: #05 CPU; #19 Classy Chassis; 64KB static RAM; a #43 2 port serial card & cables; #68 DMA Controller; all necessary cables, power regulators, and filter plates; GMXBUG monitor; FLEX; and OS-9 GMX I. You can software select either FLEX or OS-9. The OS-9 Editor, Assembler, Debugger, BASIC-09, and RUNB are also included.

#49 with dual 40 track DSDD drives \$4398.49
 #49 with dual 80 track DSDD drives \$4698.49
 #49 with #88 8" Dual Drive Disk System \$5998.49
 #49 with #90 19MB Winchester subsystems & one 80 track DSDD drive \$7398.49

#39 128KB SYSTEM includes: #05 CPU; #19 Classy Chassis; 128KB of static RAM; a #43 2 port serial card & cables; #68 DMA Controller; all necessary cables, power regulators, and filter plates; GMXBUG monitor; FLEX; and OS-9 GMX II. You can software select either FLEX or OS-9. The OS-9 Editor, Assembler, Debugger, BASIC-09, and RUNB, and GMX-VOISK for FLEX are included.

#39 with dual 40 track DSDD drives \$4998.39
 #39 with dual 80 track DSDD drives \$5298.39
 #39 with #88 8" Dual Drive Disk System \$6598.39
 #39 with #90 19MB Winchester subsystem & one 80 track DSDD drive \$7998.39

UnIFLEX, available at extra cost, requires 8" or Winchester drives. A signed license agreement with TSC is required before shipment.

You can add to any GIMIX system RAM, I/Os and other options, or substitute non-volatile RAM. GIMIX will customize to your needs.

COMING SOON: Contact GIMIX for price and availability on 40MB and 72MB Winchester (5 1/4") drives, removable pack Winchesters, 256KB static RAM boards.

All GIMIX systems are guaranteed for 2MHz operation. GIMIX systems include documentation for all boards and software in a GIMIX binder. **ALL DRIVES ARE 100% TESTED AND ALIGNED BY GIMIX.**

ALL BOARDS AND SYSTEMS ARE ASSEMBLED, BURNED-IN, AND TESTED. GOLD-PLATED BUS CONNECTORS ARE USED.

TO ORDER BY MAIL: SEND CHECK OR MONEY ORDER OR USE YOUR VISA OR MASTER CHARGE. Please allow 3 weeks for personal checks to clear. U.S. orders add \$5 handling if order is under \$200.00. Foreign orders add \$10 handling if order is under \$200.00. Foreign orders over \$200.00 will be shipped via Emery Air Freight COLLECT, and we will charge no handling. All orders must be prepaid in U.S. funds. Please note that foreign checks have been taking about 8 weeks for collection so we would advise wiring money, or checks drawn on a bank account in the U.S. Our bank is the Continental Illinois National Bank of Chicago, 231 S. LaSalle Street, Chicago, IL 60693, account #73-32033. Visa or Master Charge also accepted.

EXPORT MODELS: ADD \$30 FOR 50HZ. POWER SUPPLIES.

GIMIX, Inc. reserves the right to change pricing, terms, and product specifications at any time without further notice.

ALL PRICES ARE F.O.B. CHICAGO

Choose from GIMIX' wide variety of system components.

The **GIMIX CLASSY CHASSIS #19** consists of a heavyweight aluminum cabinet, constant voltage ferro-resonant power supply, and SS50 Mother board with baud rate generator board
 \$1398.19
 Triple Disk regulator card and cables \$88.22 Baud rate generator card \$88.93
 Missing cycle detector \$38.23 Filter plates \$14.92
 Back panel connector plates (specify) \$8.60 50 Hz. option \$30.00

MEMORIES (GIMIX uses only Static RAM)

#67 64KB NMOS STATIC RAM board \$478.67
 #64 64KB CMOS STATIC RAM board w/battery back-up \$568.64
 #34 8K PROM board \$98.34
 #32 16 socket PROM/ROM/RAM board \$238.32

I/O Boards (see above for Intelligent I/Os)

#41 Single port serial, RS232/20ma. current loop \$88.41
 #43 2 port serial, RS232 \$128.43
 #46 8 port serial, RS232 \$318.46
 #42 2 port parallel \$88.42
 #4 8 port parallel \$198.4
 #50 serial, RS232, RS422, RS423 \$244.50
 #52 SSDA serial, RS232, RS422, RS423 \$254.2
 #64 ADLC serial, RS232, RS422, RS423 \$268.54
 Each cable with connectors for back panel mounting (specify board) \$24.95

DISK CONTROLLERS

#68 DMA (featured in all systems above) \$588.68
 #28 dbl. dens. programmed I/O (5" drives only) \$298.28
 #58 single dens. programmed I/O (5" and/or 8" drives) \$226.58
 #48 same as #58 but for 5" drives only \$198.48
 Cable sets: 8" with Back Panel connector \$29.25
 for two 8" external drives \$44.26
 for two 5" drives \$34.96

SOFTWARE: GIMIX exclusive versions of OS-9/GMX I, II, III & FLEX are for GIMIX hardware only. All versions of OS-9 require the #68 controller. When ordered with any controller, FLEX is \$30.00

GMXBUG PROMs and manual

Boot or Video boot PROM \$30.00 UNIFLEX boot PROM \$60.00
 OS-9 GMX I \$200.00 OS-9 GMX II \$500.00
 Editor \$125.00 Assembler \$125.00
 BASIC-09 \$200.00 RUNB \$100.00

DISK DRIVES FOR GIMIX SYSTEMS -- complete with cables and power regulators.

5" DSDD 40 track 2 for \$900.00
 5" DSDD 80 track 2 for \$1308.00
 #88" Dual 8" DSDD drives, cabinet, power supply, & cables \$2898.88
 Cabinet only \$848.18 220V 50Hz. Option, add \$30.00
 Filter plate \$14.83 Cable for 2 drives \$44.82
 Cable for 4 drives \$87.84 Cable for cabinet to mainframe \$4.81

WINCHESTER SUBSYSTEMS: for use only in GIMIX systems with #68

DMA controller.
 #90: includes one 19MB drive, interface, and Software \$588.90
 #91: includes two 19MB drives, interface and Software \$528.91
 Contact GIMIX for price and availability of other forthcoming subsystems.

OTHER BOARDS

#75 GHOST 80X24 VIDEO BOARD \$398.76
 #66 50 pin Protoboards \$56.66 #33 30 pin Protoboards \$38.33
 #03 6800 CPU \$224.03
 #06 6800 CPU with timers \$288.06 Baud rate option, add \$30.00
 #08 RELAY DRIVER (board, bracket, transformer, and 31 relays) \$1128.08
 #86 - #08 (board, bracket, transformer, without relays) \$538.86
 #85 OPTD board \$348.85
 WINDRUSH EPROM PROGRAMMER \$37.00
 3" Binder 12.00 2" Binder \$9.00

GIMIX DOES NOT GUARANTEE PERFORMANCE OF ANY GIMIX SYSTEMS, BOARDS OR SOFTWARE WHEN USED WITH OTHER MANUFACTURERS' PRODUCT.

DON'T SEE IT??? ASK! OUR BROCHURE HAS MORE COMPLETE DESCRIPTIONS AND SPECS. PHONE OR WRITE TODAY FOR YOUR COPY.

BASIC-09 and OS-9 are trademarks of Motorola Systems Corp. and MOTOROLA, Inc. FLEX and UnIFLEX are trademarks of Technical Systems Corporation, Inc. GIMIX, GHOST, GMX, CLASSY CHASSIS, are trademarks of GIMIX, Inc.

GIMIX inc.

1337 WEST 37th PLACE • CHICAGO, ILLINOIS 60609
 (312) 927-5510 • TWX 910-221-4055



Microware presents 4 new OS-9 software packages.

1 LEVEL II PRINT SPOOLING SYSTEM

This versatile package gives your OS-9 Level Two System a complete print spooling management capability for time-sharing applications. Features of the spooling system are:

- Handles up to seven independent spooling devices and queues with "print on first available device" feature.
- Prints large block header pages between listings with date, time, user name and job name.
- Multiple listing copy option.
- Complete forms change capability for each job and device.
- Prints formatted or unformatted listings.
- Status command displays print queues and status.
- User can kill or change priority of queued jobs.

Available only for OS-9 Level Two Systems.

Suggested List Price: \$150.00 Manual Only: \$15.00

2 RMA RELOCATABLE MACRO ASSEMBLER

At last — a full feature relocatable macro assembler and linkage editor for OS-9. RMA permits sections of assembly language programs to be independently assembled to "relocatable object files". The linkage editor takes any number of program sections and/or library sections and combines them into a single executable OS-9 memory module. Global data (including indexed and direct addressing modes) and program references are automatically resolved in the process. The macro facility permits commonly used statement sequences to be defined, then used within the program with appropriate parameter substitution. RMA also supports conditional assembly and library source files.

Suggested List Price: \$200.00 Manual Only: \$20.00

3 OS-9 FILE HANDLER TOOLBOX

Introducing a special toolbox for OS-9 users who do a lot of file manipulation! A collection of 12 useful OS-9 command

programs; Most can be used as "filters" using OS-9 pipeline facilities. Included are:

D — unformatted directory listing with "wild card" matching
Compress — does character compression on text files.
Expand — restores a "compressed" file to the original state.
Split — breaks a file into smaller files.
Space — indents lines with optional spacing between lines.
Code — decodes any key on a keyboard to hex.
Qsort — quick sort for small files, directories, etc.
Pr — versatile formatted file printing utility.
Tr — transliterates text pattern to substitution pattern.
Grep — searches file for a pattern and prints matching lines.
Xmode — same "lmode" except changes are made to the device descriptor.
Count — counts words, lines, or characters within a text file.

Suggested List Price \$85.00

4 ENTERTAINMENT PACK I

A collection of games and other interesting programs that are not only entertaining but serve as good instructional examples of Basic09 programming techniques. All programs include complete Basic09 source files and can be easily edited to run on standard alphanumeric or graphics terminals.

BlkPak — A Vegas-rules blackjack game.
Cik — graphical display of a wall clock on your terminal.
Dogs — Greyhound racing with simulated graphics.
Eliza — Basic09 version of the famous artificial intelligence simulation of natural language dialogue with a psychiatrist.
Haiku — Program that creates original "haiku" prose.
Quest — a mini-"Adventure" game.
Rats — find your way out of a computer-generated maze — from a rat's point of view.
Towers — a graphical display of the solution to the "Tower of Hanoi" puzzle.

Suggested List Price: \$85.00



MICROWARE.

Microware Systems Corporation
P.O. Box 4865 • Des Moines, IA 50304
515-279-8844 • Telex 910-520-2535

OS-9 and Basic09 are trademarks of Microware and Microsoft.
Units is a trademark of Bell Labs.

FLEX™ USER NOTES THE 6800-6809 BOOK

By: Ronald W. Anderson
As published in 68 MICRO JOURNAL™

The publishers of 68 MICRO JOURNAL are proud to announce the publication of Ron Anderson's **FLEX USER NOTES**, in book form. This popular monthly column has been a regular feature in 68 MICRO JOURNAL SINCE 1979. It has earned the respect of thousands of 68 MICRO JOURNAL readers over the years. In fact, Ron's column has been described as the 'Bible' for 68XX users, by some of the world's leading microprocessor professionals. Now all his columns are being published, in whole, as the most needed and popular 68XX book available. Over the years Ron's column has been one of the most popular in 68 MICRO JOURNAL. And of course 68 MICRO JOURNAL is the most popular 68XX magazine published.

As a SPECIAL BONUS all the source listing in the book will be available on disk for the low price of: FLEX™ format only — 5" \$12.95 — 8" \$16.95 plus \$2.50 shipping and handling, if ordered with the book. If ordered separately the price of the disks will be: 5" \$17.95 — 8" \$19.95 plus \$2.50 shipping and handling.

Listed below are a few of the TEXT files included in the book and on diskette.

All TEXT files in the book are on the disks.

LOGO.C1
MEMOVE.C1
DUMP.C1
SUBTEST.C1
TERMEN.C2
M.C2
PRINT.C3
MODEM.C2
SCIPKG.C1
U.C4
PRINT.C4
SET.C5
SETBAS1.C5

File load program to offset memory — ASM PIC
Memory move program — ASM PIC
Printer dump program — uses LOGO — ASM PIC
Simulation of 6800 code to 6809, show differences — ASM
Modem input to disk (or other port input to disk) — ASM
Output a file to modem (or another port) — ASM
Parallel (enhanced) printer driver — ASM
TTL output to CRT and modem (or other port) — ASM
Scientific math routines — PASCAL
Mini-monitor, disk resident, many useful functions — ASM
Parallel printer driver, without PFLAG — ASM
Set printer modes — ASM
Set printer modes — A-BASIC
(And many more)

**Over 30 TEXT files included in ASM (assembler) — PASCAL — PIC (position independent code) TSC BASIC-C, etc.

NOTE: .C1, .C2, etc. = Chapter 1, Chapter 2, etc.

This will be a limited run and we cannot guarantee that supplies will last long. Order now for early delivery.

Foreign Orders Add \$4.50 S/H

Softcover — Large Format

Book only: **\$7.95** + \$2.50 S/H

With disk: 5" **\$20.90** + \$2.50 S/H

With disk: 8" **\$22.90** + \$2.50 S/H

See your local S50 dealer/bookstore or order direct from:

Computer Publishing Inc.
5900 Cassandra Smith Rd.
Hixson, TN 37343
(615) 842-4601

™FLEX is a trademark of Technical Systems Consultants



WE CAN EMULATE THIS CHIP BETTER THAN ITS MAKER.

Advanced Digital's Test Station Integrates and Debugs Your System Faster.

Introducing Advanced Digital's new 4009B Test Station. The test station that combines emulation and logic analysis into one system. Until now, most emulators operated in real-time by restraining the target system. Not with Advanced Digital. Our 4009B Test Station is totally transparent. Totally real-time. We meet or exceed manufacturer's timing specifications. This is particularly important when emulating interrupt intensive systems and accommodating DMA operation.

Universal operation enables you to emulate 6809 and 6809E series microprocessors without making software or hardware changes. Up to 128K of memory overlay, four hardware triggers and 2K deep of trace help you pinpoint problems faster. A line-by-line assembler and a

powerful trace disassembler are standard. They enable you to view software and make changes without returning to the development system.

MORE THAN AN EMULATOR

As a "digital engineering test station," Advanced Digital's 4009B gives you a synergistic approach to integration problems. Two microprocessors control the system. One is dedicated to emulation while the second microprocessor supervises the 32 channel logic analyzer and all system menus. This design lets you emulate and perform all system functions simultaneously.

Discover the most powerful diagnostic tool available for hardware and software integration.

Call Advanced Digital Technology for a free demonstration.



4009B TEST STATION



**Advanced
Digital
Technology**

13400 Northup Way, #27
Bellevue, WA 98005
(206) 643-2382

TMP (Total Management Planning) OS-9 SOFTWARE

THESE OFF-THE-SHELF APPLICATION PACKAGES HAVE CHANGED THE PICTURE
FOR PROFESSIONAL SOFTWARE USERS

Here are the comprehensively supported OS-9 application software packages that discerning professionals are selecting today. Each one runs stand-alone or in combination with other TMP packages.

TMP/CALC is a new-generation "Electronic Spreadsheet" package that goes a quantum leap beyond ordinary Calc software. Superior speed, and extra features like Dynamic Calculations and Dynamic Overlay, mean you can change a cell anywhere on the spreadsheet and get automatic updates wherever they belong. No more "wrap-around" since output to the printer is formatted. "Help" screens prompt without erasing data, and the rows and columns on any worksheet are limited only by available memory.

ONLY \$250

TMP/Manager is a structured database manager, and is the ultimate driving force behind any total Management/Marketing Planning system. Its built-in select/sort/merge module — and optimized machine code — makes it super-fast, and provides unusually nimble data manipulation characteristics. Other outstanding features are detailed prompting, easy identification of data fields, and an excellent security system.

ONLY \$500

TMP FreeForm fills the void between a structured DBMS and word-processing. It's an "electronic index card" package with an endless list of valuable applications such as parts cataloging, look-up inventories and listings of any kind. Type anything on up to 32,000 electronic "cards" of up to 9 pages each that reside in "file drawers" (databases). Up to 13 key-words can call each page.

ONLY \$150

TMP/Front-End integrates any or all of the packages described above, and integrates TMP with word processors, BASIC and other high-level languages. NO CHARGE with TMP/Manager



TMP PACKAGES ARE EXCEPTIONALLY FRIENDLY. EACH COMES WITH MULTI-MEDIA TRAINING AND OUTSTANDING DOCUMENTATION

AUDIO CASSETTE TRAINING TAPES are provided with each TMP software package. These crisp, professionally produced training aids take a new user

through "hands-on" exercises that instantly build confidence, and minimize reference to manuals.

CLEAR DOCUMENTATION is specially prepared for each TMP package. Remarkably readable because they are conversational in style and logically organized, this series of manuals further assures quick user comfort and productivity with TMP software. Includes sections on how to best utilize the system; integrating user-written software with TMP packages, and other bonus topics.

VIDEOTAPE PROGRAMS are available to TMP Dealers. They dramatically take the viewer into real business environments for a close look at how a TMP software package is utilized. Video programs can be personalized for specific organizations. Programs may be specified for most popular tape formats.

THE BOTTOM-LINE IS THIS: Software is only as effective as its training and documentation. TMP packages are incomparable in both areas.

RUN TMP IN A SMOKE SIGNAL CHIEFTAIN™ COMPUTER TO EXPERIENCE THE KIND OF SPEED AND POWER YOU CAN EASILY AFFORD TODAY



Endurance-Certified Chieftains are consistently among the fastest computers in CPU and I/O speeds according to the widely respected BENCHMARK REPORTS. That, combined with almost legendary reliability, makes Chieftains the logical choice in computers using OS-9 operating systems.

Configurations range from floppy-disk systems to multi-user, multi-tasking Winchester hard disk systems with tape back-up; performance approaching mainframes at prices you can live with.

TMP software packages for most OS-9 systems and Chieftain computers are offered exclusively from Smoke Signal. Inquiries from non-Smoke Signal dealers are invited. Call (213) 889-9340.

TMP T.M.



SMOKE SIGNAL
Chieftain Computers

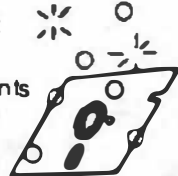
31336 Via Colinas • Westlake Village, CA 91362
(213) 889-9340

BUBBLE MEMORY

HIGH RELIABILITY STORAGE
FOR THE SS -50 BUS

DISK BUB provides 128K bubble memory replacing a disk drive.

- no moving parts to wear out
- direct boot capability
- withstands harsh environments such as dust, heat, vibration
- faster than standard drives
- utility software supplied



Diskbub plugs into the 30 pin I/O bus.
Flex09 drivers and boot rom provided.
Available for only

\$845⁰⁰



2457 Wehrle Drive, B-68
Buffalo, New York 14221
Phone (716) 631-3011

Dealer Inquiries Welcome

Business Software For The COLOR COMPUTER

WITH FLEX* & XBASIC

Data Base Manager

Part I	\$ 49.95
Part II*	\$ 49.95
Church Contributions	\$ 49.95
Balanced Billing System	\$ 49.95
Single Entry General Ledger	\$ 49.95

Integrated Business Software*

*available for Color Computer only

Accounts Receivable	\$ 99.95
Accounts Payable	\$ 99.95
General Ledger	\$189.00
Inventory 2	\$ 69.00
Payroll	\$ 99.95
64K memory upgrade including installation	\$ 79.00



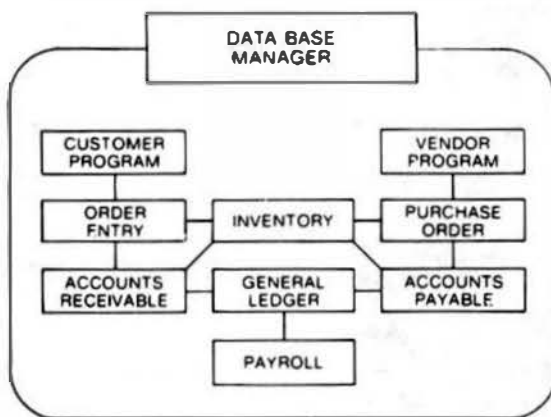
Call or Write for free catalogue

2457 Wehrle Dr., D-80, Buffalo, NY 14221
PHONE (716) 631-3011

Dealer Inquires Welcome

417

FLEX* and UniFLEX* software for the 6809



Integrated Business Programs

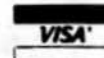
	FLEX	UniFLEX
Accounts Payable	\$295	\$395
Accounts Receivable	\$295	\$395
General Ledger	\$295	\$395
Inventory 2	\$295	\$395
Payroll	\$295	\$395
Data Base Manager	\$350	\$450
Word Processing		
Software	\$295	\$395
WP Menu		\$150
P Control		\$150



*FLEX & UniFLEX are Trademarks of Technical Systems

2457 Wehrle Drive, D-68, Buffalo, NY 14221
PHONE (716) 631-3011

Dealer Inquires Welcome



Color Micro Journal™

5900 Cassandra Smith Rd. ★

Hixson, TN. 37343

COLOR MICRO JOURNAL™ Is A Monthly
Tabloid Publication for Color Computer USERS!

★★

COLOR MICRO JOURNAL™ is a Magazine FOR
Color Computer Users BY Color Computer Users. Col-
umns on various compatible Operating Systems, Lan-
guages, Uses (Bulletin Boards, Clubs, using the RS
BASIC, and so on), etc.

★★★

Programs - Games - Reviews - Education - Hardware -
Software - New Product Announcements - Books

★★★★

Get the MOST from your COLOR COMPUTER
without being an Engineer.

DON'T MISS A SINGLE ISSUE

Subscription Rate of only \$16.50 a Year!!!

★★★★★

COLOR MICRO JOURNAL™ published by the
ONLY pure 68xx, INTERNATIONAL Computer Maga-
zine. '68 Micro Journal has provided coverage for over
FIVE Years. We KNOW the Color Computer, the Soft-
ware (both FUN and WORK) that IS and CAN BE run on
it. We KNOW the products that ARE, CAN BE, or
WILL BE used on the Color Computer.

For Ordering Subscriptions By Phone

Call 1-800-338-6800



Color Micro Journal

Limited Time Charter Rates

USA - \$16.50 per year. Canada & Mexico - \$23.00 per year

Surface Foreign - \$28.00 per year. Airmail Foreign - \$52.00 per year

*Color Micro Journal is a trademark of Computer Publishing Inc.

.....
Yes! Start my copy of Color Micro Journal coming as soon as possible!

Name _____

Address _____

City _____

State _____

Zip _____

☐ Visa ☐ Master Card ☐ Check or Money Order Enclosed

Card # _____

Exp/Date _____

Flex User Notes

Ronald W. Anderson
3540 Sturbridge Court
Ann Arbor, MI 48105

Compile It or Assemble It?

I'm going to use this forum to make public a little private debate that many of you might not only find interesting, but be stirred to respond to. (I know. Winston Churchill said once that a preposition is something you should never end a sentence with.) Before I get to the heart of the matter, I have to give you a little background. A couple years ago, I wrote a book on programming in Pascal. As one of the chapters, I wanted to tackle a non-trivial program as an example of something more complex than a "say hello" program. I ended up writing a text formatter that would accept text with embedded commands and produce a nice output with features such as centering of titles, right justification of short lines, fill and justification of paragraphs, etc.

The example program worked out reasonably well, and the book was published with version 9 or so of the program. I kept adding features to it, and it ended up recently at version 17. When I got involved in the PL/9 language, I decided that it would be shorter and faster if I translated it to PL/9. As a side benefit, it would be a good way to learn the intricacies of PL/9 and to find a bug or two if they were to be found. The translation was accomplished, and the output code was about half as large as the original. It ran several times faster than the original P-code Pascal version.

At that point I decided to implement some of the fancy features of the Epson MX-80 with Graffiti, and I added the capability of switching print modes within a line. I decided to see if South East Media would want to market it. I also decided to send three copies to various people for comments. I had been working with Dan Farnsworth of Palm Beach Software, testing his SPELLB (sold by S. E. MEDIA, see ad this issue) and music software, and I had really ripped up some of the music compiler to add the capability of loudness control. I figured I owed Dan a chance to be critical of something of mine, and he certainly was. The result of my implementing his suggestions (a dozen or so) was a greatly improved text formatter. One of his suggestions removed an awkward way of breaking the fill mode at the end of a paragraph, that had even tripped me up once in a while.

Patience, the plot is thickening rapidly. I had written the formatter as an example of a Pascal program. When I rewrote it in PL/9 it was reduced to about 4K of object file. I was quite pleased with its speed. It will dump formatted text to a CRT about as fast as the LIST command does. It can certainly keep up with most dot matrix printers and specialty "daisy wheel" types. One of Dan's comments was something like "Now that you have gone to all the trouble of writing this, why not code it in Assembler?" That hit a nerve somewhere in my constitution, and a friendly debate started. I allowed as how 4K was certainly small enough for a text formatter, and that it was certainly adequately fast. The high level language source is easier to maintain, takes less time to list, etc. In fact, from my first use of Assembler, I had been looking forward to the day when compilers became capable enough and efficient enough so that I wouldn't have to write assembler code anymore. I certainly got enough practice at Assembler for a couple of years before such a compiler appeared.

Dan's reply to that was something like "You still don't understand how I program in Assembler." My reply to him was "I understand quite well how you program in assembler, but you don't understand how the rest of us program in assembler!" Some folks are by nature organized. Some get things done. Usually people who fall into the extreme of one of those categories don't fit the other at all. I classify myself as a doer who is almost completely disorganized. Dan is an exception. He is well organized AND he gets things done. Dan, over the years, has developed a set of STANDARD subroutines for doing just about anything. He claims (and I don't doubt) that he can put together a 400 line Assembler program and have it running in a couple hours (that's about 8 pages of source).

I don't doubt Dan's claim for an instant (for DAN, that is). My problem is one of discipline. Every time I used a "standard" subroutine, I would want to change it to fit the present application exactly. (I know that for a fact because I've been there). Well, the discussion didn't end there, and hasn't as of this writing. I realize that there are areas where Assembler should be used, but I maintain that those areas are getting smaller as compilers become more efficient. Dan has written a business software package all in assembler with standard subroutines for math and formatting of numerical output. It beats the pants off of anything written in BASIC in the way of an accounting system. The programs are short so you don't spend all day waiting for another program to load to do another function. This seems to me to be a very good example of a case where Assembler really makes a difference. Of course if you have a hard disk, the load time for a BASIC program is not objectionable, but I'm talking about SMALL computers.

"C" was developed as a system programming language, that is one in which a programmer could implement an operating system. In fact the famous UNIX is written in "C". The "designers" of "C" must have known what they were doing, because benchmarks coded in "C" always seem to run fast. The "C" compilers end up at the top of the speed list. A look at the output code of a good "C" compiler (that is the assembler source code), or at the disassembled object code of the PL/9 compiler, would convince most any good Assembler programmer that he is not going to be able to do MUCH better. While I'll agree that there may be a factor of two to four possible in going to Assembler, it is not the 50 or 100 factor that existed a few years ago. I was able to code the Prime benchmark (the one in BYTE) in assembler so it runs in about 3.3 seconds on a 2 MHz 6809. Two "C" compilers end PL/9 ran it in 10, 10, and 12 seconds respectively. That is only a factor of 3 or 3 1/2 for those compilers.

There are those who cry "but look at how many bytes of object code you generated." With our 64K memories available for around \$300, I wonder if that is a valid objection. That text formatter mentioned above is 4K in PL/9. In assembler it might be 2K or possibly 1.5K. Does it really matter? The text file is read from a disk and processed a line at a time. There is no need to buffer the whole text in memory. In most of our systems, we don't have multi tasking and multiple users. For most of us, programming 15 pages of PL/9 or "C" is a very much smaller job than writing 60 pages of Assembler! Another programmer can read and understand our high level code much more quickly than he can understand our Assembler code. A "C" programmer can program a Z-80, a DEC PDP-11, a 6809, a 6502, or whatever. A 6809 Assembler programming expert can only program a 6809. Of course he can learn to program those others in reasonable time, just as I can learn to run a new editor or a different operating system.

Well, that's my side of the argument. I'll send a copy of this to Dan Farnsworth and let him write a rebuttal. I'd sure like to know what you readers think. What applications demand Assembler programs, and where are high level language programs better. (You can read better as more economical, more maintainable, portable, or any way you like).

Who's the Joker?

HEY Don Williams, The July issue of '68' contains a letter from Bill Ball complaining about the degree of reduction that some of the things in '68' undergo before printing. Some JOKER (I can't imagine who) printed that letter so small that I had to get a magnifier to be able to make out most of it, and I do wear glasses for close-up work. Maybe Bill, you ought to wear glasses too!

Don's Note: Ron that is the size we received it. I really don't remember ever printing any article or anything else quite that 'small'. However, we print a lot of 'stuff' other magazines would not bother with. I do it because I feel the need or benefit, to my readers, exceeds the requirement of large, solid, dark or otherwise good copy material. Maybe that is why over 95% of our readers resubscribe. We ain't fancy but we gotta lot of good stuff!

DMW

"C" Senior

I thought of labeling this bit "Confessions of a hard headed Swede". I'm not totally unfamiliar with "C", but I have never really gotten deeply into using it. (I'm going to save some typing and omit the quotes around C for the remainder of this). I decided to try to convert my

scientific functions package that I had written in Pascal some time ago, and recently upgraded thanks to the efforts of Matt Scudlere in '68 in the April Issue. I've recently been working on the PL9 version of that package in cooperation with the folks at Windrush, so I had the PL9 version to start with.

Generally when I try out a new language, I first get a feel for a few simple programs. I thought in this case, that I was past that point. The next step is to try to translate something significant from another language, and I set off to do just that with the scientific functions. My first attempt at a test program and the implementation of the SINE and COSINE function produced a screenful of errors. Oh, yes, the syntax of C is a bit different for an "if-then" statement. The condition must be in parentheses. Oops, still an error, undefined symbol "then". Ah, so, C doesn't want the word then. It expects only "if (condition) statement;". Well, that one went away, and when I put the parentheses into all the other "if" statements and deleted the "then", most of the errors disappeared.

I had another peculiar error that pointed out a difference between C and most of the others (I had forgotten this one). Not equal in C is represented by "!=" rather than the more common "<>". Again, when the error message told me that there was an error on that line, it was obvious.

I still had one very stubborn error in two places in the program. I had remembered to declare my functions "float sin(angle) etc.", and thought that was all I had to do to insure return of a float value from the function. A careful reading of Kernighan and Ritchie (at 1:30 AM) indicated that I have to declare the function in the calling program too, right along with the local variable declarations. "float angle, number, sin(), cos();" seemed to take care of those last compiler trapped errors.

NOTE: I later found out that those function declarations are only necessary if the function call is a "forward reference", sort of analogous to the FORWARD call in Pascal. That means that calls to library functions don't need the function declarations since (or if) the #include for the library file is before the main() function in the C program.

Now to run the thing. Gee, it actually ran, with only one problem. I put in 45 degrees as the angle, and it returned 0.000000 for the sine and 0.000000 for the cosine.

Well, after a couple hours next morning (Saturday) of adding print statements and recompiling, I found the error of errors for neophyte C programmers. Of course I knew better! Only with the evidence pointing at one or two lines did the problem become obvious. Actually, there are two problems. They reinforce each other. The first is the "uncommon" use of "==" for an equality test. The second is the "understood" !=0 (ie. not equal 0) that is built into a test. You can get away with such things as "do <statement> while (count--);". The "while" condition tests for the value of the argument, and anything BUT 0 is considered TRUE. Therefore, initializing an integer "count" to 10 will make that loop run 11 times. (The trailing -- causes the value to be decremented AFTER the test, so the loop runs for count ==0 as well as 10 through 1.)

Now put that little feature (which I like a great deal because it allows shortcuts in programming) together with the == feature and you get disaster. I had intended to test my variable "angle" in the "sin" function, and return the value 0 if the angle is 0. There is no use going through all the calculations to get approximately zero when exactly zero is known to be the answer. Well, I ended up with the following:

```
if (angle = 0.0) return 0.0;
/* continued calculation of sine */
```

Do you see the disaster? Since C allows an assignment statement within the condition. I unconditionally set angle = 0.0. "if (angle = 0.0) is always FALSE BECAUSE IT FORCES angle = 0, and of course 0 is FALSE. Naturally the function didn't "return 0.0" because the condition was false. It DID set angle to 0, so the following calculation of the sine of the angle DID return 0. Changing the (angle = 0.0) to (angle == 0.0) changed an unwanted assignment statement to the intended TEST, and suddenly sin returned an answer (of course it was incorrect). I found a simple and unrelated error that I corrected easily to fix that problem.

I've ordered "The C Primer", reviewed by Norm Commo in the July '68 Micro Journal. Perhaps a more organized and step by step approach will make learning C a bit less a chore. I managed to get a square root function going after some considerable trouble on Saturday morning after finding the problem with sin and cos. (The cos function simply adds PI/2 to the angle and calls the sin function). The sqrt function involves repeating an approximation calculation, each time getting a more accurate answer, until the difference between two successive answers is smaller than some limit. Since the two numbers being compared get closer together with each pass through the calculation, but either may be larger than the other, an absolute value function is needed. No problem, thought I, just use "while (abs((result - guess)/result) > 1.0e-6)" to keep the loop going until the error is small enough.

Oops again.. abs is a defined function, but it is an integer function. Now all I had to do is define absf as a function that returns a float, and declare it in "sqrt" as I indicated above. That worked fine too about 8 compilies later after I had found and fixed all my little errors.

That one down, I added the "atan" function (arctangent). By now, I had made most of the easy mistakes, and so didn't fall into any serious traps. I tried declaring some "boolean" values as char, but of course, the library had defined TRUE and FALSE as integers. That one was easy. Now "atan" worked fine but again... The Wrong Answer! This time a careful look revealed that I had mistyped one of the series coefficients.

I worked on into the evening adding each new function to the package and testing it. No more peculiar errors came to light, just a very hard to find typographical error in an array index and that "==" trap again. This time I knew what the problem would be when the result came out as though a variable had the value zero, and found it instantly.

Norm Commo, this probably has you in stitches by now! What's my point? First, the entertainment value of this might make it interesting reading. Second, C is a bit more "intricate" than some of the other languages. You can make an error and still have a perfectly legal statement (as in the == case above). Of course when you do this, the compiler has no way of knowing that you didn't intend to do what you did, so it doesn't report an error. This saga of C is not intended to be critical, just a factual relating of what happens when this programmer tackles a new language. This reinforces my belief that there is no substitute for USING a language to become proficient in it. I had read about C for some time, and had used it previously for some simple test programs, but hadn't REALLY gotten into anything deep.

OS9 USER NOTES

By: Peter Dibble
517 Goler House
Rochester, NY 14620

The OS-9 User Seminar

On August 12 the OS-9 User Seminar opened rather slowly as I and a few other people stood in line in front of the exhibit hall on the third floor of the Des Moines Marriott. We watched as various Microware staff struggled to get a Radio Shack computer (running OS-9 of course) interfaced with a television. When I got into the hall, I was surprised at the number of exhibitors. I have always thought of the OS-9 community as about the size of a large family -- there were 24 booths listed in the exhibitor guide. I had a ball wandering through the hall, meeting people I have only known through phone conversations, and seeing some exciting hardware.

Several of the exhibitors were showing machines that used OS-9 as a process control environment. One booth sported a rack of equipment that would have been more at home next to an assembly line.

Smoke Signal broadcasting had a video tape rig showing a movie of a military-looking man. I remember a bugle and a lot of strutting up and down, but I just can't remember what he was talking about; I think he was promoting the TMP package. Smoke Signal had a compact SS50 based machine that I have never seen before.

There were a couple of Japanese engineers demonstrating Fujitsu FM-7 and FM-11 computers. Very well done. I wish they were available in this country. A particularly nice feature of the software on the Fujitsu machines was split-screen support. I saw them editing on one part of the screen while two other sections displayed moving graphics ... all running at the same time.

Tano was showing a Dragon computer, imported (I believe) from England. The Dragon is a small, inexpensive computer with color graphics and OS-9 Level One. I only saw it playing games, but it does that pretty well.

Privac was showing the graphics board that I have been coveting for months now. It looks even better in reality than it appears in an advertisement. There was a program running almost continuously that demonstrated the board. Figures and characters would appear, disappear, rotate, and float across the screen. I had always wondered how well the Privac board was supported under OS-9; it turns out that OS-9 is the operating system they use. The demo program was written in Basic09.

Wires from the Glimx booth seemed to spread all over the hall. The OS-9 User Group, JBM Group, and Frank Hogg Labs all were borrowing computer services from Glimx. Perhaps to demonstrate the tireless ability of GMX-111 to spew characters out on many terminals, unused terminals were kept busy listing strange programs. Whenever I walked by, one of the terminals at the FHL booth was listing a COBOL program. At the Glimx booth I met the engineer responsible for my hardware (who is also the president and the service manager of Glimx). He thinks Glimx hardware should move in about the same direction I want it to go. If things go well, there should be some terrific new hardware coming out sometime in the indefinite future.

The JBM Group is hard for me to characterize. They had some utilities that sounded good except that they were written at least partly in Basic09. The thing that upset and fascinated me was that they have a sort which they claim runs very fast. They claim to have compared their sort to a standard disk-based merge sort, and come out significantly better. Either the algorithm used by the program they compared theirs to was not the best available in the literature, or their claim may have to be placed in the same class as perpetual motion machines. The man who invented their sorting algorithm wasn't there for me to ask about the details of his method, and I had no way to check their figures, so I will continue to view their sort with skepticism; however, even if it is only an average sort, its manual documents a fine general sorting program of a type which is much needed by serious OS-9 users. They had several other packages including a set of Basic09 subroutines for ISAM file handling that sounded much less exotic, but interesting.

There were, of course, many exhibitors I haven't mentioned (for example Microware's own booth), but I don't intend to make this column into a walking tour of the exhibit hall.

Friday there was plenty of time to look around. Saturday and Sunday were so busy there was barely time to eat. Microware filled most of the weekend

with classes, presentations, and "roundtables" ranging from OS-9 and Basic09 Features, which covered things like the Basic09 editor, to the OS-9 Roundtable, which gave us a chance to interrogate the parents of OS-9 about its workings. In the evenings a few of the exhibitors ran "hospitality suites" which gave some of us an excuse to stay up late and talk about our computers.

Saturday night there was a meeting of the OS-9 User Group. The User Group is having some troubles which seem to stem mostly from having only a few members spread over a wide geographical area. We elected officers for the next year: Dale Puckett (President), myself (Vice President), George Dorner (Treasurer), and Tom Murphy (Secretary). We are respectively responsible for the Software Exchange Committee, the Membership Committee, the Communications Committee, and the By Laws Committee.

Monday those of us who were still left around went off to Microware's offices. I had a chance to discuss some of the difficulties I am having with OS-9 and C with the appropriate people, and discovered that those programmers are seriously crowded. They desperately need to make the move to a larger facility that they have been planning.

Shell Commands

The shell is a program that interprets command lines and does what is called for. The full Unix shell is a programming language in itself. The OS-9 shell is only a subset of the Unix shell, but it has enough flexibility to be useful. The first thing to learn about the shell is how to use the built-in shell commands. The `chd`, `chx`, `ex`, `kill`, `w`, and `setpr` commands are built into the shell. The shell commands are used to control the environment of the programs that are run by the shell.

I use the `chd` command, which is the command which changes the working data directory, more than any other shell command. The working data directory is the directory which will be used for most files you read or write without specifying a directory in the file name. It is usually much better to change the working directory than to explicitly include directories in file names so I frequently change directories as I change from one task to another. It is a rare day when I use the `chx` command, the command which changes the working execution directory, even once. I imagine that someone with a smaller system disk than mine would use the `chx` command much more frequently than I do because OS-9 remembers where the working directories are on disk, and needs to be reset with `chd` and `chx` commands when a disk is changed. If you forget to change directories when you change disks, OS-9 will give you a nasty message next time you try to use the directory. I have never gotten into trouble by forgetting, but it is not wise to trust an operating system too far.

The `ex` command should be classed as an advanced command. It replaces the shell with another program. Replacing the shell is certainly a good thing to be able to do, especially for users with smaller systems, but it can have disconcerting results -- mainly that when the program ends, the shell won't be there.

The rest of the shell commands are primarily useful for those who run programs concurrently. You can instruct the shell to start a program running, then give you another shell prompt by putting an `&` after the command on the command line:

OS9: `dir >p&` would list the files in the data directory on the printer while you run other programs.

If you run programs concurrently, the kill, setpr, and w commands will be useful. The kill command should be used about the way you use the quit control key (usually <CNTRL>Q or <CNTRL>E). The quit key only works on the last program to do I/O to the terminal, the kill command works on any program. The setpr command is used to control the way the computer's resources are divided up. The higher the priority of a program, the larger a share of the computer it will get, and the faster it will run. A program's (or, more properly, process's) priority can be anywhere in the range 1 to 255. The w command causes the shell to wait for a child process to finish. That means that the shell won't prompt for another command until a program that was started by it terminates. The main use of this command is to recover from the mistake of running a program that does I/O to the terminal in background. The usefulness of the w command can be appreciated by trying the following experiment:

```
OS9: dir x& Now try to get some useful work
done ... when you are disgusted with the screwy
behavior of your terminal, type w at the OS9 prompt:
OS9: w
```

There is one particularly nice feature of the shell which is, so far as I know, undocumented. If you run a program like the assembler with its output directed somewhere other than the terminal, then decide that you would like to run another program at the same time, you can cut the assembly loose from the shell with the interrupt control key (usually <CNTRL>C). The interrupt control key will usually terminate the program which most recently did I/O to the terminal, but, if the program in control of the terminal (the assembler in this case) doesn't do any I/O to the terminal at all, it won't kill it. Instead, the shell sees the interrupt, and converts the program in control of the terminal to a concurrent program.

A Logical Device Driver

This column is an experiment with a new format. There is a demand for information for new OS-9 users, but I have also heard requests for more advanced discussions. In this column I am trying to include something for everyone. What follows may be of general interest, but, for an inexperienced computer user, it may be heavy going.

Several months back I started a project whose objective was to find a way to give OS-9 a terminal-independent way to control CRTs. I have a special device driver which does just what is called for, but it is built around Microware's ACIA source. I may be able to get permission from Microware to publish the modified driver, but I would rather not have the terminal mapping tied that closely to the computer's I/O port. Not every computer uses an ACIA chip for its serial interface, and my special driver only works with ACIA serial interface chips. What is needed is a virtual, or logical, device that can insulate the terminal mapping code from the physical interface.

The idea of a logical device driver has many applications beyond a terminal-independent interface. At the User Seminar I spent some time talking to the engineers from the Fujitsu booth. They wanted my opinion of a proposal to make logical devices a part of OS-9 in order to allow the system drive to have a consistent name regardless of the type of hardware being used. This would make it easier to write programs that referenced files on the system drive. A logical device can certainly do this. It would be possible to set up a logical disk with some obvious name like SYS, and have it know the name of the physical drive being used as the system drive: D0, H0, or whatever.

The idea can be extended even further. There is no compelling reason why the logical device should refer to exactly one physical drive. The logical device could refer to several physical devices or just a part of one.

Some possible uses of the concept are:

A disk drive with associated cache storage.

A neat, and fairly easy way to support split screen terminal displays with each section of the screen treated as a separate terminal.

A gateway to a network.

A way to associate a printer with a terminal for screen dumps.

A terminal-independent interface.

The device driver I have included with the column is a logical SCF device driver for a Level Two system. A RBF driver, or a driver for Level One would be somewhat different, but only the details would need to be changed. The driver, which I named VCIA, doesn't do anything at all except waste time. It is a skeleton for something interesting to be built on.

Starting from the top, let's go through the interesting parts of the program. A logical driver must look just like a real driver to the system, so it must have the type Driver+Object, and it must have a byte after the normal module header reflecting the modes in which the driver can be used. This one says it is good for update, it might be a good idea to add execution. The storage required by a device driver is called the device static storage, it is allocated, and partly initialized by the file manager, in this case SCF. The file manager uses the first section of the device static storage, the storage reserved for it with the "org V.SCF."

Performance is crucial at this level. Every character read from or written to the terminal will pass through this module so even a small improvement in efficiency is good. Normal good coding practice is still important, but the priorities shift somewhat. SCF will branch to entry, entry+3, entry+6 ... depending on what service it wants. The normal convention is to put a list of lbra instructions here, but a bra instruction is a little faster, so I used them and padded them to three bytes each with nops (which are never executed in this context).

The INIT call must find the physical device driver and set up the proper environment for it. The physical driver, which I call P.D., is found and mapped into the address space with a F\$LINK call. This is a bit tricky. In Level Two, the module is linked into the address space belonging to the process doing the link. The device driver uses the process number of the process that opened it, but it runs in the system address space. I had to fool the operating system into thinking I was running under the system process number by playing with the pointers in the system direct page. If this were Level One, I think I could have simply used a F\$LINK call without all the fussing around.

The real device driver is going to need its own device static storage, so the logical device driver plays SCF for a moment and gets the amount of storage P.D. needs. I use the Level Two system memory request, Level One users can probably use the Level One analog. The address of the memory must be

saved for future calls, and I save the size for convenience.

One never knows when SCF will change its part of the static storage, so before each call information from VCIA's static storage must be copied to P.D.'s static storage, and after each call information must be copied back from P.D.'s static storage, to VCIA's.

The INIT call, and each other call, basically changes from VCIA's static storage to P.D.'s and calls the appropriate entry in P.D.

The TERM entry is responsible for cleaning up as the device is closed. After calling P.D. to allow it to close down the physical device, it frees the device static memory that INIT allocated for P.D., and unlinks P.D. It is worth noting that TERM is the mirror image of INIT.

The device descriptor I use for VCIA is called VTERM. Since SCF thinks VTERM is a real device, its device descriptor is important. Even the address of the port that VTERM uses is important. If you keep things as simple as I did, the logical device driver will map everything including the information from the device descriptor directly to the physical device driver, but, if you want to support something like split screen, you will have to give each logical terminal a different port address, or SCF will know they all are referring to the same device, and get in the way.

Fortunately this virtual driver works with GIMIX I/O processors. This is pure luck because GIMIX doesn't publish enough information about their driver for me to design an interface for it. Let's consider this a gentle push for Richard Don at GIMIX to release more information about his proprietary software.

It should be possible to move a good deal less data back and forth between VCIA's and P.D.'s static storage than I do, but I played it safe in spite of the large cost. It would be good to try to find some fields that don't need moving so time could be saved by not moving them.

This module demonstrates that, although OS-9 Level One is compatible with Level Two for user programs, it is not compatible for system modules. This shouldn't be a surprise, but it is something to be cautious about. In many cases all that needs to be changed is an entry in a definitions file, but if you try to run VCIA as it stands in a Level One system, the best you can hope for is that it will give an error code and quit.

Debugging code in system state is not something I will do if I have a choice. The debugger won't work on modules that need to run in system state, and debugging code that writes out helpful messages as a program runs doesn't make sense in a device driver module. If the driver doesn't work what do you write to? I debugged this module by using its return code. If you have VCIA set carry before it returns to SCF, the program that is trying to use it will get the value that was in the B accumulator when VCIA returned to SCF. This is a slow way to learn things, but it works.

One final point: It is expensive, but otherwise impeccable technique to pile logical device on logical device. VCIA has no way of knowing whether the device it believes controls the physical device is real or logical.

Microware OS-9 Assembler 2.1 09/10/85 20:42:45
vcia - Virtual (logical) device driver

```

00001      nam      vcia
00002      tti      Virtual (logical) device driver
00003
00004      * This module should be used as a SCF (Sequential)
00005      * Character File device driver. It doesn't
00006      * drive any specific device, but, rather, calls
00007      * a physical device driver, such as ACIA, to
00008      * deal with the physical device.
00009      * Possible uses:
00010      *   o Mapping various terminals to a standard
00011      *   o Implementing windows.
00012      *   o Linking a P/A and an ACIA to provide
00013      *     switchable printing of terminal output.
00014
00015      * As it stands this module is a dummy. It passes
00016      * all calls through with minimum interference.
00017
00018      * The INIT call must set up the environment for
00019      * the device driver before passing the call on.
00020
00021      * Read, write, getstat, and setstat can probably
00022      * get away with less than they do, but all
00023      * variables are copped between control blocks
00024      * to ensure that this module is transparent.
00025
00026      * The TERM call must release memory allocated for
00027      * the physical driver before returning.
00028
00029      -----
00030      [FP]      use /B0/DEFS/defslst
00031      ENDC
00032      type      set      Driver/Object
00033      Revs      set      RevEntol
00034      0000 07C000F7  rdb      Vciolen,Name,Type,Revs,Entry,RevSize
00035      0000 C3      fcb      Updat.      Driver can be used for updated (read + write)
00036      000E 5a4349C1  nam      fcs      /VCIA/
00037      0012 01      fcb      l      edition number
00038      0013 414349C1  pblen     fcn      /ACIA/
00039
00040      ; Device static storage for this virtual driver
00041      ;
00042      0 0010      org      V.SCF      room for SCF variables
00043      0 0010      pblen     rdb      2      Pointer to Physical Driver's Header
00044      0 001F      pblen     rdb      2      Pointer to Physical Driver's Entry
00045      0 0021      pblen     rdb      2      Pointer to Physical Driver's Static Memory
00046      0 0023      pblen     rdb      2      amount of memory allocated for PB static mem
00047      0 0025      revsize   equ
00048
00049      ; Block of entry points
00050      ;
00051      0017      Entry
00052      0017 2010      bra      Init
00053      0019 12      nop
00054      001A 20A3      bra      Read
00055      001C 12      nop
00056      001D 20A6      bra      Write
00057      001F 12      nop
00058      0020 20A3      bra      GetStat
00059      0022 12      nop
00060      0023 20A6      bra      PutStat
00061      0025 12      nop
00062      0026 160999      lbra      Fore
00063
00064      ;
00065      ; last time the driver it will use a physical device driver,
00066      ; and of, states and initializes its static storage
00067      ; Passed:
00068      ;   u      points to static storage
00069      ;   f      points to device descriptor
00070      ;
00071      0029      Init
00072      0029 1460      push     T,B
00073
00074      ;
00075      ; adjust process number to system process
00076      ; so the line will be into the system address space.
00077      ;
00078      002B BC50      ldd      B.Proc
00079      002B 1466      push     B
00080      002B BC46      ldd      B.SysProc
00081      0031 0030      std      B.Proc
00082
00083      0033 0aE1      ldd      dType      driver module type
00084      0033 2080FFBd  ldd      Pblen,PCB  point 1 of the name of the P Driver
00085      0039 103F00      cbr      FCB,0
00086      003C 1366      push     B
00087      003E 0030      add     B.Proc
00088      0040 10250009      lbra      Error
00089
00090
00091

```



```

00090 0044 4E62      ldx 2,S      copy address of device static store to x
00091 0044 EF9B1B   stw PModH,1 save P.B.'s Module Number address
00092 0049 10AFB01F sty PModH,1 save P.B.'s Entry address
00093
00094 004B EC4B      ldx #PModH,U memory requirement of P Driver
00095 004F 10572B   ORY FESModH request system memory
00096 0052 10250077 ldx Error1
00097 0056 E09B23   stw PModH,1 save amount of memory allocated
00098 005A E09B21   stw PModH,1 save pointer to memory

```

00100

```

00100 005E C41B      ldx #V.SCF length to save
00101
00102 * At this point I points at vcia's static storage
00103 * U points to P.B.'s static storage
00104
00105 *****
00106 * Move the entire set part of the device static storage
00107 * into P.B.'s static store.
00108
00109 005E LMoveE
00110 005E 5A        decb
00111 005F 2B06      bcs #Three
00112 0061 04B5      jda B,1
00113 0063 07C5      vld B,0
00114 0065 20F7      bra #MoveE go around loop again
00115 0067
00116 0067 352B      puls Y      but leave U in the stack
00117 *****
00118 * U points to P.B.'s static storage
00119 * Y points to the device descriptor
00120
00121 0069 AEBB1F     ldx PModH,1
00122 006C ADB4      jwr #Init,1 do P B init
00123
00124 006E 1F31      tfr U,1
00125 0070 3540      puls U
00126 0072 3404      puls B
00127 *****
00128 * now I points at PB static store
00129 * U points at vcia static store
00130
00131 0074 B062      bsr #MapIn
00132
00133 0076 3584      puls B,PC      return to SCF

```

```

00134
00135 * SCF needs to see any changes the physical device
00136 * driver makes to V.Paus, for V.Err1.
00137 * I points at P.B.'s static storage.
00138 * U points at vcia static storage.
00139
00140 0078 #MapIn
00141 0078 E60B      jdb v.Paus,1
00142 007A E74B      stb v.Paus,1
00143 007C E60E      jdb v.Err,1
00144 007E E74E      stb v.Err,1
00145 0080 39        rts

```

```

00146 0081 #Read
00147 0081 E603      ldx #V.Paus
00148 0083 205A      bra Common

```

```

00149 0085 #SetStat
00150 0085 C609      ldx #V.Paus,1
00151 0087 200E      bra Common

```

```

00152 0089 #PutStat
00153 0089 C60C      ldx #V.Paus,1
00154 008B 2002      bra Common

```

```

00155 008D #Write
00156 008D C606      ldx #V.Paus,1

```

```

00157
00158 * Code used by all entries except INIT
00159 * Passed B -- offset from P.B.'s entry point for this op.
00160
00161 *****
00162 008F #Common
00163 008F 7446      puls B,U
00164 0091 AECB21     ldx PModH,U
00165
00166 * The physical device driver needs to save
00167 * V.LPNC, and V.BUSY explict to it each time it is
00168 * called.
00169 * At this point B points at vcia static storage.
00170 * I points at P.B.'s static storage.
00171
00171 0094 EC45      ldx V.LPNC,0
00172 0096 E043      stw V.LPNC,1
00173 0098 EC45      ldx V.LPNC+2,U

```

```

00174 0098 E045      stw V.LPNC+2,U
00175 009C EC47      ldx V.LPNC+0,B line
00176 009E E047      stw V.LPNC+0,1
00177 00A0 EC49      ldx V.LPNC+0,U bus2
00178 00A2 E049      stw V.LPNC+0,1
00179 00A4 EC4B      ldx V.LPNC+0,U Intr
00180 00A6 E04B      stw V.LPNC+0,1
00181 00A8 EC4D      ldx V.LPNC+10,U PCh
00182 00AA E04D      stw V.LPNC+10,1
00183 00AC EC4F      ldx V.LPNC+12,U sdn
00184 00AE E04F      stw V.LPNC+12,1

```

```

00185
00186 * The P.B. requires Y, and A to be as they were
00187 * when vcia was called. A and Y have been
00188 * been disturbed.
00189 * U must point to P.B.'s static storage.
00190 * When Common was called,
00191 * B pointed to the offset from P.B.'s entry point
00192 * that we should jump to.
00193
00194 00B0 AECB1F     ldx PModH,U
00195 00B2 EECB21     ldx PModH,U
00196 00B4 2506      puls B      recover offset in P.B. and A from stack
00197 00B6 30B5      jwr B,1

```

```

00198 00B8 1F31      tfr U,1      point I at PB static storage
00199 00BA 1540      puls U      recover pointer to vcia static
00200 00BC 1494      puls B      save return code
00201 00BE 1506      bsr #MapIn      (I points to PB static, U points to vcia stati
00202 00C0 1506      puls B,PC      return to SCF

```

```

00203 00C4 #Term
00204 00C4 140F      ldx #B.Paus
00205 00C6 30C7      bsr Common      call PB
00206 00C8 1445      puls CC,U      CC and error 0 from P.B.
00207
00208 00CA 1F31      tfr U,1
00209 00CC EECB21     ldx PModH,1      address of PB static storage
00210 00CE EECB23     ldx PModH,1      size of memory
00211 00D0 13F2F9     ORY FESModH      return memory
00212 00D2 2519      bcs Error2
00213
00214 00D4 EECB1B     ldx PModH,1      address of module header
00215 *****
00216 * adjust process descriptor to system process
00217
00218 00D6 BC50      ldx B,Proc
00219 00D8 3406      puls B      save B
00220 00DA BC4A      ldx B,StrProc
00221 00DC B050      stw B,Proc

```

```

00222 00DE 103F02     ORY FRMLimit      unlink PB
00223 00E0 3506      puls B      recover old Process Descriptor
00224 00E2 B050      stw B,Proc      and restore it in place
00225 00E4 2505      bcs Error2

```

```

00226 00E6 35C5      puls CC,B,U,PC      return to SCF

```

```

00227 00E8 #Error1
00228 00E8 3560      puls Y,U
00229 00EA 39        rts
00230 00EC #Error2
00231 00EC 3262      ldx 2,S      clear CC and B off stack
00232 00EE 35C0      puls U,PC      return to SCF

```

```

00233 00F4 1051A3     ENDB
00234 00F7 #VciaLum
00235 00F7 0000      mov vcia
00236 00F9 0000      lli vcia Device Descriptor Module
00237 *****
00238 * Vcia device descriptor module
00239
00240 00F1 #Type
00241 00F1 07C000SE    mov DEVC OBJECT
00242 00F3 03         lcb #PModH,1
00243 00F5 0FE024      lcb 00F,0E0,024 part 2 side B
00244 00F7 1B         lrb #PModH,1
00245 00F9 0000      mov #
00246
00247 0000 options 0000
00248 0012 00        B.Class      lcb 00      SCF
00249 0013 00        B.Mac       lcb 00      #PModH
00250 0014 01        B.Bus       lcb 01      Backspace echo (BS) (SP) (DE)
00251 0015 00        B.Bin       lcb 00      delete clear line
00252 0016 01        B.Echo      lcb 01      echo on
00253 0017 01        B.Alt       lcb 01      auto line feed on
00254 0018 00        B.Pan       lcb 00      on end of line mails
00255 0019 00        B.Pan       lcb 00      end of page pause off
00256 001A 18        B.Pan       lcb 24      Lines per page
00257 001B 00        B.Pan       lcb 00      Backspace char

```

00257	001C 18	0.Del	fcb	24	delete line char
00258	001B 09	0.Eor	fcb	13	end of record char
00259	001E 10	0.Eof	fcb	27	end of file
00260	0016 04	0.OPR	fcb	04	reprint last line
00261	0020 01	0.Dup	fcb	01	duplicate last line
00262	0021 17	0.Psc	fcb	23	pause character
00263	0022 03	0.Int	fcb	03	keyboard interrupt char
00264	0023 15	0.Out	fcb	15	(end) CTRL+M 17 keyboard abort char
00265	0024 08	0.Dsr	fcb	08	backspace echo char
00266	0025 07	0.Ovi	fcb	07	<null>
00267	0026 15	0.Par	fcb	15	device initialization value
00268	0027 00	0.Bsu	fcb	00	band rate
00269	0028 002F	0.D2p	fcb	10000000	attached device name
00270	0029 00	0.Dm	fcb	000	
00271	002A 00	0.Ioff	fcb	000	
00272	002E 0000	0.Sta	fcb	0	offset to states routine
00273	002E 00	0.Err	fcb	00	
00274	001B	0.Dltn	eqs	4-OptStrt	
00275	002F 36544352	forName	fcb	/VTEB/	Device Name
00276	0034 5343Ca	FNName	fcb	/SCF/	File Manager name
00277	0037 564349C1	DRName	fcb	/VCIA/	Driver name
00278	003B 007713	ENDC	ENDC		module CMC
00279	003E	forName	eqs	4	
00280		fcb			

```

00000 error(s)
00000 warning(s)
00135 00309 program bytes generated
00008 00000 data bytes allocated
0233E 00022 bytes used for symbols

```

BT-1 GRAPHICS BOARD REVIEW

David Hanon
Ringgold, GA

High resolution graphics is finally available for SS-50 bus users.

I heard about the BT-1 before it was advertised and placed my order right away. The board arrived near the date promised and was of very high quality construction. The bus connectors were gold plated and the board was double sided with plated through holes and was solder masked and had silk screened designators for all components and option switches.

When the BT-1 arrived I was anxious to put it into operation; but, to my disappointment there was no manual shipped with it. Even so, I was able to put it into operation in "text mode" and display the character set by doing some trial and error poking to the I/O registers. At first sight I was quite impressed at the resolution of the characters but somewhat disturbed with jitter on the small characters. This problem was explained a couple of days later when the manual arrived under separate cover. The manual explained that for best display of the small character set and graphics displays a long persistence monitor is recommended. A quick trip to a local computer store solved this problem. I found the "Apple Monitor III" green phosphor monitor to have long persistence and after trying it on the BT-1 found it to be an ideal monitor for it. It completely stopped the jitter and gave a very crisp and readable display.

The manual contains source listings for print drivers to operate both in FLEX and in OS-9. Also included are several example programs for learning operation of the board and a program to dump graphics displays to an Epson printer.

The BT-1 has two modes of operation, "smart" terminal mode and graphics mode. The board powers up in terminal mode with 48 lines of 83 characters each. I will list some of the commands available in terminal mode.

esc "0,1,2,3" selects either the supplied character set or one of three optional user defined character sets in ram.

16

esc "=" X,Y sets the cursor to X,Y location on the screen.

esc "B" binary load sequence.

esc "C" starts a ram character set load.

esc "G" switches to graphics mode.

esc "J" returns a value from one of the eight A-D converters on the board.

esc "U" Jump to user program on BT-1 board.

Other commands perform erases and set single and double character heights and widths.

Now we come to what the BT-1 is all about. High resolution graphics! The BT-1 uses STD syntax commands for all graphics functions. Although the screen resolution is 512 horizontal pixels by 480 vertical, coordinates are specified as 2048 by 1920. This allows compatibility with software written for other high resolution displays and it also gives a display size of approximately 100 coordinate points per cm on a 12 inch monitor.

The graphics commands are quite comprehensive and I will describe most of the important ones.

AXIS DRAW

Draw an X or Y axis starting at any specified point and with any specified number of tick marks.

BROKEN LINE

Set the segment lengths for broken lines for special effects.

Color

Set color to black or white.

DRAW LINE

Draw a line from any specified coordinate pair on the screen to any other coordinate pair. Will either draw white or black depending on the color flag.

HOME

Home the graphics cursor.

LINE FLAG

Enables or disables broken line feature.

MOVE RELATIVE

Moves the graphics cursor + or - in the X or Y direction relative to the present position.

POSITION

Moves the graphics cursor to the specified X,Y position.

SIZE TEXT

Sets the graphics mode character size to one of 15 different sizes.

TILT TEXT

Print text normally, up vertically, down vertically, or up-side-down.

VECTOR DRAW

Draws lines in the same manner as MOVE RELATIVE moves the cursor.

WRITE TEXT

Writes text on the graphics screen in the size specified in the size flag and can optionally write in italics.

ZIP UP

Switches back to terminal mode.

That just about sums up the commands available on the BT-1. Now, let's discuss some aspects of the operation of the board.

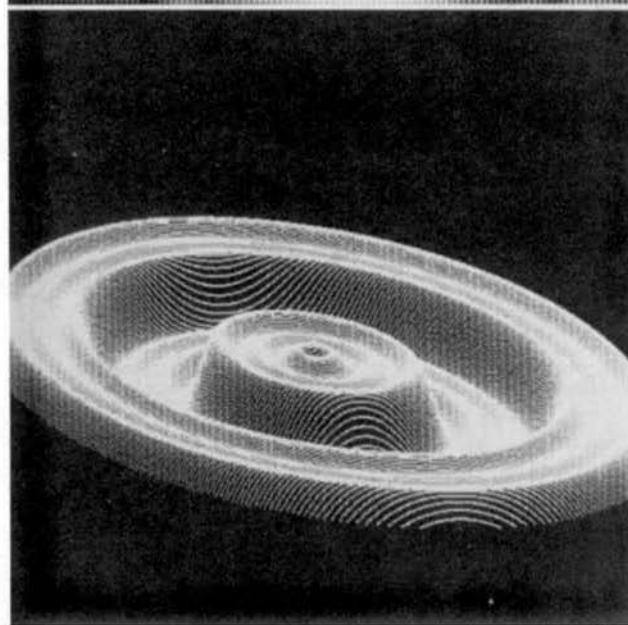
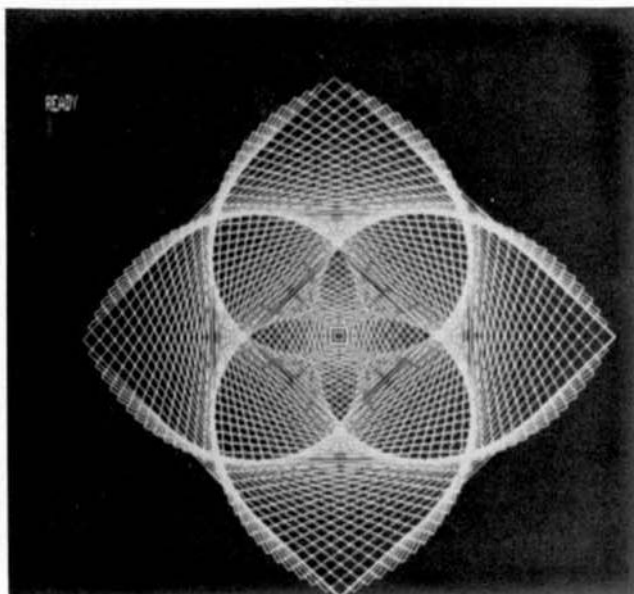
The BT-1 is addressed at \$E380 as shipped and occupies 4 bytes of address space. It may be moved to any desired address and supports extended addressing.

Communication with the BT-1 is via 2 ports, the first of which is a data register and the second a status register. Data is sent to or received from the data register and busy or data ready handshaking is done by reading or writing to the status register.

The Electro Screen operates with a 6809 processor on board and 32 K of display memory, a 2 K scratch pad RAM, 6 K of EPROM firmware, and 2 more sockets for additional user defined character sets or user programs.

You may download your own programs to the board and use it as a parallel processor if desired. The display is controlled by the "old reliable" 6845 CRT controller. This allows much flexibility in making changes to the display format under software control. For example, I mentioned that with a short persistence monitor there was some flicker on the display, this can be eliminated by a poke to the control registers of the 6845 which changes the display format to two pages of 24 lines and 83 characters per line. There is no scan interlace in this mode and it gives a perfectly flicker free display on any standard monitor.

I have spent many hours programming the BT-1 and I find it very rewarding to work with. The high level commands available make it easy to program complex patterns. I recently modified a program which was written for the "APPLE" computer to run on this board. As I entered the program I suddenly realized that 14 lines of complex code were used just to plot coordinate axes on the screen. It only took 1 simple line to do the same thing on the BT-1.



Another interest of mine is displaying video images via computer. I set out to write a program which could display pictures on the screen. The result was a basic program which arranged the screen into 128 by 120 pixels each consisting of 16 individual dots which could be turned on or off to simulate 16 levels of gray. The results were quite satisfying. I have included a sample of this display as well as some other graphics displays hoping Don will find room to include them with the review.

Still another use I have found for the BT-1 involves a project which I have needed to do for some time. I have a need to measure the temperature of a nearby stream versus ambient temperature. This was easily accomplished by connecting 2 temperature sensors to 2 of the 8 analog to digital converter inputs on the BT-1. The result was a real time temperature indicator which could display the temperature on the screen or make a long term graph on the display and even dump it to a printer for a permanent record.

As you have deduced by now, I am extremely enthusiastic about this device, and rightly so. The

hardware is very professional, and the manual is unusually clear and easy to use (something unusual in the computing field).

I have only one minor complaint. The SET CURSOR command requires that you add \$20 to the values that you send to the board. This is no real problem if you keep it in mind but is a minor irritation. Maybe this will be fixed in later versions of the firmware. One other suggestion might be to include a command to set one pixel. Presently, it requires a "P" command and a "D" command to turn on one pixel. All other commands seem adequate and other functions such as circles and other geometric figures may be drawn with simple user written subroutines.

I believe that anyone with the need for graphics in their business or just for entertainment will be more than satisfied with this product.

To order or for more information on the BT-1

Jason Yoon Private, Inc 3711 S. George Mason Dr.
Fall Church, Va. 22041 (703) 671-3900

NEW COPY UTILITY

A NEW COPY UTILITY FOR FLEX 1,2 and 9

By: Leo Taylor
Bruno Puglia
109 Twin Brook Rd.
Hamden, Conn. 06514

GETTING STARTED

COPY can be run by any FLEX user without reading this manual. How can such a claim be possible? COPY is compatible with the command syntax of the copy program supplied with FLEX, thus if you can use FLEX you can use this program. You are urged to assemble the source program provided and start using COPY even before you read past this page of the manual. Though you will be using only a part of COPY's capability you will have many hidden advantages such as error checking, alphabetizing, and date retention. When you have time to read the remainder of this manual you will learn how to use the many options supported by COPY.

ASSEMBLING THE SOURCE

COPY is provided as a source file that can be assembled by the FLEX assembler for 6800 or 6809 processors. The only change that need be made for CPU is one character in the line:

FLEX EQU \$A000

Locate this line using your favorite editor and confirm that it matches the FLEX you are using. The FLEX starting address should be \$A000 for 6800 or \$C000 for 6809. Note only ONE CHARACTER need be changed, you do not need to do a global change of a lot of addresses. This is the only change that must be made, there are optional changes that will be explained later. COPY can now be assembled and the object should be named COPY.CMD. You may want to retain your old copy program under a different name (such as COPY.OLD) until you gain confidence in the new program. It won't be long before you are convinced that this program can do everything the old copy program can.

INITIAL SURPRISES

COPY will accept commands in the same format as the original copy, but you will immediately notice differences in how the program performs. There will be changes in timing, such as pauses when sorting the directory and half as many head loads per file copied. You will notice files are copied in alphabetical order. The original creation date of the file being copied will be used for the destination file. If your disks have bad files you may get some unfamiliar error messages. COPY has a lot of error checking which will uncover problems you may not be aware of. The most common error is:

BAD DATE: FILENAME.EXT SIZE DATE
ALLOW BAD DATES (Y/N)?

Later in the manual there will be a section on correcting file dates, for now you can override the bad date by pressing "Y" when the prompt appears.

TERMINOLOGY

Throughout this manual these terms will be used often:

COPY	the new program
OLD COPY	the program provided with FLEX
SOURCE	the file being copied from (any type of file)
DESTINATION	the file being written to
OPTION LETTERS	command line letters used to enable options
MATCH LIST	list of starting characters (see below)

It will be assumed that you are familiar with the syntax of the COPY command provided with FLEX. All FLEX users know that the old copy program supports a match string list for copying multiple files that match the letters in the list. The command:

COPY 0 1 AB X .CM

will copy from drive zero to drive one all files that start with "AB" or "X" or have extensions that start with ".CM". This is the most common syntax you will use with COPY.

Many FLEX users do not know that the old copy program supports a useful syntax for copying a single file. If you want to copy just the file "P.CMD" without including other command files starting with "P" you will find a match list copy unsuitable. The command:

COPY 1.P.CMD 2

will copy ONLY ONE FILE from drive one to drive two. A variation of this single file copy is used to copy one file and change the name:

COPY 0.FLEX2.SYS 1.FLEXNEW.SYS

In entering any COPY command either a comma or a space can be used to separate parameters.

OPTION LETTERS

There are over a dozen option letters supported in COPY which makes it very powerful and a bit overwhelming. It is important to know that you do not need to understand all of the options to use COPY effectively; in fact there are some options you may not WANT to use. If you can't remember the letter for the option you need you can type "COPY" without any parameters and the program will display a HELP list of option letters with a short explanation of each. Option letters are included on the command line after the word COPY and before any drive number or file name:

COPY LDN 0,1 A.TXT

Any number of letters may be used in any order. If an unimplemented letter is used COPY will display the HELP list of options. Options are positive; that is using the letter will enable the option. This contrasts with the confusion of the FLEX assembler which has some negative options. You may recall that assembler options +NS means YES include line numbers but NO do not include symbol table. Much of the confusion with this otherwise excellent assembler is due to the ambiguous method of handling option letters. It was decided to avoid ambiguity by having all option letters enable the desired option. If a letter is accidentally used twice the option is still enabled.

DEFAULT OPTIONS

COPY has default options which will take effect whenever there are no option letters on the command line. The most common default option will be "A" which enables sorting of the source directory before files are copied. COPY has provisions for changing the default options in the assembler source program for those users who want other options defaulted. For example one could default only option "M" and COPY will emulate the old copy by using the current FLEX date on the destination file. If the user must change the default options there is a table near the start of COPY which contains the default option bytes. Each byte is FCB 0 if the option is not defaulted and FCB 1 if it is defaulted. If the user needs to verify

which options are defaulted, typing COPY without parameters will display an example followed by all the options. The defaulted options will be marked with "YES":

Example: CO Y ADNL 0,1 .CMD A

```
YES A copy in Alphabetical order
NO C allow Corrupt files to be copied
NO O copy files with newer Date
NO E delete Existing destination file
NO F copy by File number (alpha not allowed)
NO K Kill duplicate file on source
NO L List files without copying
NO N copy files Not on destination
NO O turn Off defaults
NO P Prompt before copying file
NO R Recover from track-sector
NO S makes Second copy of file (.CPY ext)
NO U Use current FLEX date
NO W Wait for disk change
NO Z Zap source file after copying
```

EXPLANATIONS OF OPTION LETTERS

A COPY IN ALPHABETICAL ORDER

The "A" option enables a sort subroutine that will alphabetize the source directory before files are selected to be copied. Most users will have this as a default option so the files will be in order when cataloged. If you enter any option on the command line all defaults are turned off. If option "A" was defaulted and you want to alphabetize you must add "A" to the options you enter.

C ALLOW CORRUPT FILES TO BE COPIED

The "C" option will enable you to copy a file that is damaged by a CRC error or record sequence error. This is a slightly dangerous option which should only be used if you don't have an alternate copy of a file.

D COPY FILES WITH A NEWER DATE

The "D" option will find files that are on both disks and compare their creation dates. If the source file is newer it will be copied as a replacement for the older destination file.

E DELETE EXISTING DESTINATION FILE

The "E" option is used when you want to replace a file on the destination disk. This option will suppress the prompt for whether you wish to delete the existing file. It will often be used along with "D" to update a disk with newer versions of programs.

F COPY BY FILE NUMBER

The "F" option changes copy's parameters from a match string list to a list of file numbers. Follow the drive numbers with a list of file numbers for those files that you want to copy. File numbers can be found with a utility such as DIR.CMD. A group of files can be specified as a starting and ending number separated by a dash. The command "COPY F 0,1 5 13-18 9" will copy file five, files thirteen through eighteen, and file nine.

K KILL DUPLICATE FILE ON SOURCE

The "K" option is VERY dangerous. This command isn't really a copy, rather it uses the directory compare routines to delete files from the source disk that appear on the destination. This allows you to clear off extra copies of programs not needed on the source disk. It operates very fast and will clear off a number of files faster than you can hit reset. As with all dangerous options it is protected with an "ARE YOU SURE" prompt. "COPY K 1,0" is most effective in killing files on drive 1 when they exist on drive 0. "COPY KD 1,0" will kill the file on drive 1 when it is older than the file on drive 0. Use COPY KDL 1,0 to preview what files will be deleted.

L LIST WITHOUT COPYING

The "L" option disables the file copy subroutine. This is used to display a list of files that WOULD have been copied if you hadn't used option "L". This can be used with other options to check disks for duplicate files, newer dates, bad files, etc.

N COPY FILES NOT ON DESTINATION

The "N" option is used to copy the files on the source disk that are not already on the destination disk. This

can be used to add all new files to a backup disk.

O TURN OFF DEFAULT OPTIONS

The "O" option is a dummy character used to turn off all default options. If you do not want any options. If used with any other option letters it has no effect.

P PROMPT BEFORE COPYING

The "P" option enables this prompt: "Prompt off (P):FLEX (F):copy (Y/N)?". You should respond with "P" if you want to continue copying without the prompt or "F" to return to FLEX or "Y" to copy this file. "N" or any other character will skip to the next file. This is useful for scanning through a disk copying only certain files. Another use is skipping down to a certain file on a disk and copying all files after that.

R RECOVER FROM TRACK-SECTOR

The "R" option is used to read a file without using the directory. If the directory of a disk has been destroyed but the user knows the file's starting address, the file can be recovered. The command "COPY R 1 2B 5 0.NEWFILE" will read from drive 1, track 2B, sector 5 until encountering an end of file or a record out of sequence. The write file extension will default to .SCR. A second use for this option is to recover a deleted file. The first sector of a deleted file can be found with the new CAT.CMD. If the file has not been over-written, COPY can recover it. Record sequence checking eliminates the restriction that the file be the last file deleted. Provision was made to start copying in the middle of a file. This is a somewhat dangerous option since it allows the user to override the FLEX File Manager. The command must be typed as shown with three numbers and a file name. No other options should be used with "R".

S MAKE SECOND COPY OF A FILE

The "S" option is used when you want two copies of a file to be written. The second copy will have the same name as the first with the extension of ".CPY". This is useful when sending a program to a friend whose drive may have difficulty reading your disks. By sending two copies there is a much higher chance that the file will be readable.

U USE CURRENT FLEX DATE

The "U" option is used when you want the destination file to have the present FLEX date rather than the date of the source file being copied. This may be useful if you know the source file has an erroneous date.

W WAIT FOR DISK CHANGE

The "W" option eliminates the need to copy COPY.CMD onto the source disk so that you may remove your system disk to insert the destination disk. When "W" is used on the command line COPY will wait for a key to be pressed before accessing any directories.

Z ZAP SOURCE FILE AFTER COPYING

The "Z" option is somewhat dangerous. It is used to delete the file from the source disk after it is copied. Essentially the file is moved from one disk to the other.

EXAMPLES

Often options can be combined to perform tasks that previously were impossible or required a separate command utility. Some examples:

COPY C 0.BAD.TXT 1 Copies one file that has a CRC error, including whatever data is readable from the bad sector. Replaces the utility RAWCOPY.CMD.

COPY DN 0,1 Updates the destination with all files from the source that are not on the destination or have an older date on the destination. Replaces the utilities COPYNEW and ARCHIVE.

COPY EZ 0.P.CMD 2 Moves file P.CMD from source to destination no matter what. Replaces the utility MV.CMD.

COPY F 2,0 1-16 Useful for copying from one large drive to smaller ones. Copies 16 files to drive zero and the remainder to drive one. This is VERY difficult to accomplish with the old copy.

COPY KD 1,0 Kill the file on drive 1 when it is older than the file on drive 0.

COPY LNA 2,1 Alphabetically lists those files on the source that are not on the destination. Replaces the utility DUP.COM and is much faster.

COPY P 0,1 Prompt before each file is copied to allow user to select which files are desired. Replaces the utility PCOPY.

COPY R 2 4 6 1.FOO Recover a file on drive 2 that does not appear in the directory. Reading will begin at track 4 sector 6 and the file will be written to drive 1 with the name FOO.SCR. Replaces the utility RECOVER.COM.

COPY W 0,1 .CMD After COPY is loaded the user is prompted with "Change disk-press key". The system disk may be removed and another disk inserted.

DA

A few option letters enable dangerous functions such as killing the original file on the source drive. These options should only be used by an "Experienced Copier" and are protected with this prompt:

dangerous option selected
are you sure (Y/N)?

If you accidentally stumble on a dangerous option type "N" and you will be returned to FLEX. With proper use these options can be quite useful.

ER

COPY has many error checks that are not found in other FLEX utilities. Many FLEX users have found that disk problems often surface when using the old copy, often with disastrous results. One error COPY traps is files with a size of zero sectors. These files usually result from pressing reset while doing file operations. Novice FLEX users do not realize that if they reboot FLEX after aborting with reset a defective file is left on the disk. When a program like the TSC EDIT.COM opens a file, FLEX creates a directory entry with zero sectors. If the program is aborted with reset or the disk is removed the zero sector file remains in the directory. Later, when the disk is copied, the defective file will result in the entire free space of the disk being copied. This copy "runaway" results in the destination disk being clobbered. Another problem which is eliminated with COPY results from a bad link in the source file causing it to intersect with another file. A third error solved by COPY is a bad link on the destination disk causing the file to overwrite the directory. A time wasting error in the old copy was it would try to copy a file that wouldn't fit in the available space on the disk. COPY will catch this error and report "file will not fit".

COPY has several of the common FLEX error messages built in. These were included since many users of two drive systems will need to remove their system disk to copy files. This results in FLEX reporting errors by number since the file ERRORS.SYS will not be found on the system drive. Also for convenience the drive associated with the error is reported along with the offending track and sector. This is useful for such errors as CRC which could occur on either the source or destination drive, and in the file or the directory.

It is important to realize that if you get an error while writing a file to the destination disk the new file may be defective. The file may appear in the directory but usually it is incomplete.

The most common error message reported by users of COPY is "DATE BAD". This occurs when the user does not enter a valid date when FLEX is booted or by failure of a hardware clock when used for setting the FLEX date. COPY will check the date on all files when it reads the disk directory and report any dates outside a reasonable range. This reduces the chance that a bad date will be passed on to the new file. There are two alternatives when the "BAD DATE" message appears. You can answer "Y" indicating that you approve of bad dates or answer "N" and not copy the file. After returning to FLEX you can re-enter COPY using option "U" which will assign the current FLEX date to the file or use the new DATE command to set the file date to the day the file was made.

SP

COPY was not written with the intention of speeding up file duplication. FLEX write speed is inherently slow and can not be improved without deviation from the normal FMS operation. On the average, COPY is comparable in

speed to the old copy. A few operations are faster such as match list searching and copying short files. A few operations are slower such as reading the directories. As with the old copy it is faster to do a single file copy than a match string copy with only one match. For example use "COPY 0.DIR.COM 2" rather than "COPY 0,2 DIR". The sorting overhead will be reduced as time passes since more of your disks will be sorted. The sort chosen for COPY is optimized for directories that are sorted except for a few new files at the end. Examination of this table will show that for most cases the differences in speed between COPY and old copy are negligible.

CO

Files copied	size	old COPY	new COPY	new sorted		
one file	5	7	8	8	+10%	+10%
entire disk	5	211	200	201	- 6%	- 5%
first file	8	10	14	19	+40%	+90%
last file	8	14	19	23	+40%	+64%
entire disk	8	1313	1245	1250	- 5%	- 5%

Five inch system disk with 32 FLEX utilities.
Eight inch data disk with 250 random 1 sector files.

ME

COPY is several times longer than the old copy, but requires less disk space than the sum of the utilities COPY eliminates (see the examples). COPY uses 4-5K bytes of memory starting in base page for variables, program, messages, and file control blocks. The directories are loaded into memory after the FCBs and require 16 bytes per file. This puts a limit on the maximum number of files at 1600 for 6800 or 2600 for a full size 6809 system. This should not be a limitation for most FLEX users. The remaining user memory (less 8800 bytes reserved for EXEC) is used as a buffer for the file being copied.

NO

Considerable care was made to determine that COPY would work on all FLEX systems. All accesses to the disks were made through the File Manager System to avoid compatibility problems with user written disk drivers. Still, it was necessary to use a few tricks in order to make the program more useful. The FMS does not provide a means for determining if a disk is protected or even if it exists without attempting to write on it. This is the FIRST thing a program should do so the user doesn't waste time answering prompts only to find the program aborts later. COPY reads the SIR (sector 3) of the destination disk then duplicates it on sector 4. This is an unused sector on all systems tested. If the disk is protected or not ready the program will exit immediately. COPY shuts off interrupts when accessing track zero to avoid problems with the spooler. COPY checks the next to the last byte (\$FE) of the SIR for an extended directory flag. This byte is set non-zero by some users to indicate the catalog has been extended to include all of track 1. If this flag is set COPY will protect track one as well as track zero. Finally the file date is inserted into the destination file FCB before the file is closed. This is done to keep the original date on the new file. None of these "tricks" have produced any problems for the FLEX users who have tested COPY; they are only mentioned in case someone has a problem with COPY that appears to be compatibility.

PR

Occasionally one may want to keep a log of the files being copied, perhaps for unattended operation. The command "P COPY 0,1" will print the file specifications as the files are copied. A duplicate of each line will appear on the terminal, along with all prompts. The command "O OUTFILE COPY 0,1" will do the same for a disk file. Since the "OUTFILE" will be open when COPY reads the directory do not be alarmed that it is reported as a BAD FILE.

CO

Many FLEX users are lax in entering the correct date when booting their system. Part of the responsibility for this lies in the old copy program which used the current FLEX date when copying a file, thus rendering the file date almost useless. With this new COPY utility the date takes on new significance since the date reflects when the file was originally created.

To use the alphabetizing feature of COPY to its full

advantage one should assign similar file names to similar files. For example, an assembler library file should have a name as close as possible to the assembler file it goes with. A documentation file could have the same name as the program it is documenting with an extension of ".DOC". This will keep the files together when copied.

CONCLUSION

The programmers who wrote COPY hope you will enjoy using this FLEX utility. COPY is written to be compatible with FLEX systems that use any 680X processor and any disc controller. There is no advantage in writing FLEX utilities that are optimized for a select few systems; this only results in dividing up the FLEX users.

Editor's Note: Again an excellent software utility from Leo Taylor and Bruno Puglia, both excellent programmers and of that rare breed that develops programs that have TRUE utility!

For those dreading the finger beating chore of typing in and debugging, the source as well as 6800 or 6809 command files, will be available from 68 MICRO JOURNAL on diskette - specify 5 or 8 inch - this is another 68 MICRO JOURNAL READER SERVICE. This service is provided for subscribers ONLY! Include your subscriber number as it is shown on your 68 MICRO JOURNAL mailing label.

PRICE
5" \$14.95
8" \$19.95

ADD: \$2.50 S/H

DMW

```

Improved COPY utility for FLEX 1,2,9
Program by Bruno Puglia and Leo Taylor

Calling format for HELP:
COPY          note no parameters

Calling format for using default options:
(Implements TSC FLEX COPY)

COPY 0,P,CMD 1      use file to new drive
COPY 1,P,CMD 0,P,BDC use file with new name
COPY 0,1,CMD P      all command files then
                    all files starting with P

Calling format for setting options:
COPY 0AP 0,1 P,CMD  clears all default options
                    then sets Date, Alphabetize
                    and Prompt flags. Copies
                    all files starting with
                    P then all .CMD files

COPY 0 2,1,111      copy all level files that
                    are not on drive 1. Plus
                    sign is ignored; included
                    only for compatibility.

Option letters:
A copy to alphabetical order
C allow Corrupt files to be copied
B copy files with same Date
E delete Existing destination file
F copy by file number
K Kill duplicate file in source
L List files without copying
H copy files not on destination
D Use Off default options
P Prompt before copying file
R Recover from track-sector
N make second copy of file
U Use current FLEX date
B Wait for disk change
Z zap source file after copying

Commas and spaces can be used interchangeably.
Default options are selected by editing the
source and will be over-ridden if options
are included in the command line.

```

```

When prompting before each file is selected COPY
will wait before file is copied and display:
Prompt off (P):FLEX (F):copy (Y/N)?

The user should respond with:
Y Yes copy this file
N No do not copy this file
P copy remaining files without Y/N Prompt
F exit to FLEX
CR and all other characters default to
no do not copy this file

*****
FLEX Address assignments:
Change the next EQUATE for 6800/6809
Use 68000 for 6800 or 6C000 for 6809

0000 FLEX EQU 6C000 START OF FLEX
0040 PCB EQU FLEX+0040

0C02 TTYEOL EQU FLEX+0C02
0C09 PAUSE EQU FLEX+0C09
0C0E DATE EQU FLEX+0C0E
0C13 GETCHP EQU FLEX+0C13
0C18 PUTC0 EQU FLEX+0C18
0C1E PUTC0 EQU FLEX+0C1E
0C21 CLASS EQU FLEX+0C21
0C24 PCOLS EQU FLEX+0C24
0C27 HATCH EQU FLEX+0C27
0C29 GETVIL EQU FLEX+0C29
0C33 SETEXT EQU FLEX+0C33
0C36 ADDR2 EQU FLEX+0C36
0C39 OUTREC EQU FLEX+0C39
0C3C OUTREI EQU FLEX+0C3C
0C3F RPTERR EQU FLEX+0C3F
0C42 BETHEI EQU FLEX+0C42
0C48 IRECE EQU FLEX+0C48

0403 FREQB EQU FLEX+0403
0406 FMS EQU FLEX+0406
MEMORY ADDRESS (LIMITS, POINTERS,
TERMS, FLAGS, WORKSPACE)

0000 ORG 000
0080 DESDIR RW 2 START DESTINATION DIRECT
0082 SRCDIR RW 2 START SOURCE DIRECT
0084 BUFFER RW 2 START BUFFER
0086 RANEND RW 2 END OF WORK AREA

0088 TEMP1 RW 2
008A TEMP2 RW 2
008C TEMP3 RW 2
008E TEMP4 RW 2
0090 TEMP5 RW 2

0092 LORAND RW 2 LOWER FILE NUMBER
0094 HIRAND RW 2 HIGHER FILE NUMBER

0096 REMAIN RW 2 SECTORS REMAINING
0098 RECORD RW 2 COUNT RECORD NUMBER
009A TRUNC RW 2 SAVE FOR BAD LTR
009C DRIVE RW 1 DRIVE IMMEDIATELY SELECTED
009E SRCDRV RW 1 SOURCE DRIVE
009F DESTDRV RW 1 DESTINATION DRIVE
00A0 DATE RW 1 1=YES -1=NO BAD DATES
00A2 BADF RW 1 0=NO 1=YES BAD FILES
00A4 BADD RW 1 1=BAD DIR ENTRY
00A6 SHUTP RW 1 0=SHORT SHUT HOME
00A8 OUTTP RW 1 GAVE OUTPUT SWITCH
00AA PAUSP RW 1 GAVE PAUSE DURING COPY
00AC SHUTP RW 1 1=SHORT GAVE ENTRY
00AE SHUTP RW 1 0=USE WATCH STREAM
00B0 FILEP RW 1 1=WRITE FILE COPY
00B2 FILEP RW 1 1=WRITE FILE OPEN
00B4 FILEP RW 1 1=READ FILE NAME
00B6 FILEP RW 1 1=COPY FILES COPIED
00B8 FILEP RW 1 1=PRINT DRIVE NUMBER
00BA FILEP RW 1 1=PRINT ADDR DETECTED

```

```

0040 DIRFLG FCB 1 1-SRC DIR BEING READ
004E PRTERR FCB 1 1-PRINT ERROR MSG/SEC
004F EXTFLG FCB 1 1-EXTENDED DIRECTORY

0080 NAMEPC FCB 4 SPACE FOR MATCH STRING
0084 NAME FCB 8
008C EXT FCB 3

```

```

0100 ORG 00100
*****
* COPY COMMAND UTILITY
*
0100 20 69 COPY BSA EXEC
0102 04 FCB 132 VERSION NUMBER

0103 2E 01 3A 01 FCB 07E,101,13A,001 3.1:1
0107 20 43 4F 50 FCB ' COPY99 June 21, 1982'
0108 50 39 39 20
010F 20 46 75 4E
0113 43 20 73 31
0117 2C 20 30 39
0118 30 32
011B 04 FCB 4

```

```

* OPTION LETTER DEFAULTS
*
* DEFAULT OPTIONS SELECTED AT ASSEMBLY
* BY EDITING THE FOLLOWING FLAGS. FCB 0
* SELECTS 'NO' AND FCB 1 SELECTS 'YES'.
*
* DEFAULT OPTION BYTES WILL BE COPIED
* TO OPTION FLAGS DURING INITIALIZATION.
*
011E 01 DEFAULT FCB 1 Alphabetize dir
011F 00 FCB 0 allow corrupt files
0120 00 FCB 0 copy by Date
0121 00 FCB 0 delete Existing
0122 00 FCB 0 copy by file number
0123 00 FCB 0 Kill source duplicate
0124 00 FCB 0 List without copy
0125 00 FCB 0 copy New files
0126 00 FCB 0 turn Off default
0127 00 FCB 0 Prompt before file
0128 00 FCB 0 Recover from tri-sec
0129 00 FCB 0 Second copy
012A 00 FCB 0 Use file date
012B 00 FCB 0 Wait for disk change
012C 00 FCB 0 Zap source file

```

* OPTION LETTERS

```

0129 41 LETTER FCB 'A
012E 43 FCB 'C
012F 44 FCB 'D
0130 45 FCB 'E
0131 46 FCB 'F
0132 48 FCB 'K
0133 4C FCB 'L
0134 4E FCB 'M
0135 4F FCB 'N
0136 50 FCB 'P
0137 52 FCB 'R
0138 53 FCB 'S
0139 55 FCB 'U
013A 57 FCB 'V
013B 5A FCB 'Z

```

* OPTION FLAGS

```

013C 00 ALPFLG FCB 0
013D 00 COMFLG FCB 0
013E 00 DIRFLG FCB 0
013F 00 EXTFLG FCB 0
0140 00 FTULFL FCB 0
0141 00 KULFLG FCB 0
0142 00 LSTFLG FCB 0
0143 00 NAMEFL FCB 0
0144 00 OPTFLG FCB 0
0145 00 PRTERR FCB 0
0146 00 RECDLG FCB 0
0147 00 SECFLG FCB 0
0148 00 UPDFLG FCB 0
0149 00 WATFLG FCB 0
014A 00 ZAPFLG FCB 0

```

000F D7SE7 EBU LETTER-DEFAULT

```

*****
* EXECUTIVE PROGRAM CALLS ROUTINES

```

* ADD SELECTS COURSE OF ACTION

```

0140 00 0104 EDCG JSR INIT SET UP
014E 00 0232 JSR PMSR READ COMMAND LINE
0151 70 0149 TST NOFLG
0154 27 04 BEQ NOHMT
0156 0E 0F07 LBI ADDRESS WAIT BEFORE COPY
0159 00 04C0 JSR ADDRESS
015C 70 0141 NOHMT TST KILLFLG CHECK DANGEROUS MOVES
015F 27 05 BEQ TRYJAP
0161 7C 0143 INC NOFLG KILL REVERSES MEH
0164 20 04 BUI DANGER

0166 70 0144 TRYJAP TST ZAPFLG
0169 26 05 BNE DANGER
016B 70 0130 TST COMFLG
016E 27 25 BEQ NOHMT
0170 0E 0E4F LBI 0144HMT OPTION SELECTED
0173 00 031E JSR PUTNIN
0176 70 0130 TST COMFLG KILL OR ZAP?
0179 26 00 BNE NOHMT
017B 0E 0EC3 LBI 00000
017E 00 0073 JSR PORTA
0181 96 98 L3AA SCHEDIN
0183 0A 30 CMAA 0'0
0185 00 C310 JSR PUTNIN OUTPUT DRIVE NUMBER
0188 70 0142 NOHMT TST LSTFLG LISTING ONLY?
018B 26 00 BNE NOHMT
018D 0E 0E40 LBI 1544HMT
0190 00 04C0 JSR ADDRESS
0193 26 62 BNE E11T3
0195 0C 40 INC PUTNIN PRINT FROM NOB OR
0197 70 0142 TST LSTFLG REPORT WITHOUT COPY?
019A 26 03 BNE NOHMT
019C 00 0009 JSR STATUS
019F 00 0330 NOHMT JSR READ2D BEST WRITE PROTECTED?
01A2 9E 82 LBI 34C10 READ BOTH DIRECTIONS
01A4 9C 84 CFI BUFFER
01A6 26 00 BNE NOHMT
01A8 0F 00 CLB PUTNIN
01AA 0E 0F47 LBI 00000
01AD 7E 04EE JMP EXCESS REPORT AND E11T

01B0 00 47 NOHMT TST SHELFL
01B2 27 03 BEQ MULT MORE THAN ONE FILE?
01B4 7E 0435 JMP SIMALE COPY ONE FILE

01B7 70 0144 MULT TST FILFLG COPY BY NUMBER?
01BA 27 03 BEQ NOHMT
01BC 7E 075A JMP REPORT

01BF 70 013C TRYJAP TST ALPFLG SORT SOURCE DIR?
01C2 27 03 BEQ NOHMT
01C4 00 04DC JSR SORT ALPHABETIZE SOURCE
01C7 00 030C NOHMT JSR SELECT SELECT OPTION FILES
01CA 70 0142 TST LSTFLG
01CD 27 05 BEQ NOHMT REPORT ONLY?
01CF 00 64 L3AA PAGEST
01D1 07 CC04 STAA PMSR PAUSE FOR REPORT
01D4 00 0470 NOHMT JSR MATCH COPY MATCH FILES
01D7 7E 04AF E11T3 JMP E11T

```

* INITIALIZE WORKSPACE

```

01DA 0E 0000 T41T LBI 00E5D10
01DB 4F 84 CLATMP CLB 0,1 CLEAN TEMPS
01DE 30 01 T41T
01E1 0C 000F CFI
01E4 26 F7 BNE CLATMP
01E6 0E 011E LBI 00000
01E9 46 84 NOHMT L3AA 0,1 HAVE DEFAULTS TO FLAGS
01EB A7 00 1E STAA 0FFHET+2,1
01EE 30 01 T41T
01F0 0C 0120 CFI
01F3 26 F4 BNE NOHMT
01F5 0E 1383 LBI 00000
01F8 9F 00 STZ D15D10
01FA 96 01 L3AA D15D10+1 FOR TESTING START ON
01FC 04 F0 L3AA 00F0 EVEN 16 BYTE ADDRESS
01FE 97 01 STAA D15D10+1
0200 04 CC20 L3AA 00000
0203 FA CC2C L3AA 00000
0206 00 00 SUBD 00
0208 97 04 STAA 00000
020A 97 07 STAA 00000+1
020C 04 CC09 L3AA 00000
020F 97 04 STAA 00000
0211 7F CC09 CLB
0214 70 CC24 TST
0217 26 00 BNE YESHMT

```

```

Leave space for EXEC etc.
end of buffer

save space
no pause during copy
output to disk?
YESHMT

```

```

0219 DE C015 L01 OUTOC=1 terminal vector
021C DE C010 CP1 OUTCH=1 printer vector?
021F 27 04 DEB NOPF1L if save no print
0221 0C AC YESPRT INC POUTFLG set print mode
0223 00 04 DEB PCOLP2 prior to next line
0225 06 CC22 NOPF1L L0AA OUTCHP save output switch
0228 07 A1 STAA OUTCHP
022A 06 FF L0AA H0FF
022C 07 CC22 STAA OUTCHP force to terminal
022F 7E C024 PCOLP2 JMP PCOLP terminal to next line

```

```

*****
*
* PARSE COMMAND LINE FOR OPTIONS, DRIVES, NAME
*
* IF BAD OPTION FOUND DISPLAY HELP LIST
*
* SIMPLE FILE COPY: FILE, I OR FILE, FILE
* MULTIPLE FILE COPY: 0,1 MATCH STRINGS
* SET OPTIONS: LIST 0,1 MATCH STRINGS
*

```

```

0232 06 CC11 PARSE L0AA LSTIRM EMPTY LINE?
0233 01 00 C0PA 000
0237 27 05 DEB HELP2
0239 01 CC02 C0PA 11YEDL
023C 26 03 DEB SAVINF
023E 7E 0401 HELP2 JMP DISPLAY HELP LIST

0241 DE CC14 SAVING L01 106701
0244 9F 90 STI 106701 SAVE POSITION
0246 00 C027 SCAM J0R WITCH SCAM LINE FOR LETTERS
0249 01 20 C0PA 014
024B 27 F4 DEB SAVING (IGNORE PLUS)
024D 01 41 C0PA 014
024F 25 04 BLD SCAM0
0251 01 74 C0PA 014 LOWER CASE
0253 25 F1 BLS SCAM0
0255 9E 90 SCAM2 L01 106701
0257 0F CC16 STI 106701 RESET POINTER
025A 01 20 C0PA 0020 SPACE FOLLOWED OPTIONS
025C 26 33 DEB NOPF1
025E DE 013C L01 MULTPL0
0261 4F 04 CLARP CLR 0,1 CLEAR ALL FLAGS
0263 30 01 INC 100
0265 0C 0143 CP1 106701+1
0268 26 F7 DEB CLARP1
026A 00 C027 OPTOP J0R WITCH SET OPTION CHNG
026B 25 14 DEB EXOPT NOT ALPHANUMERIC
026F 04 3F ANBA H0FF MAKE UPPER CASE
0271 DE 012C L01 001101-1 OPTION LETTER TABLE
0274 30 01 OPTLO2 101
0276 DE 013C CP1 MULTPL0
0279 27 C3 DEB HELP2
027B A1 04 C0PA 0,1
027D 26 F5 DEB OPTLO2
027F 4C 0F INC OPTRET,1 SET OPTION FLAG
0281 20 E7 ANBA OPTOP
0283 DE CC14 EXOPT L01 006701
0286 06 04 L0AA 0,1
0288 01 20 C0PA 0020 SPACE?
028A 27 04 DEB S0PSEP
028C 01 2C C0PA 014 COMMA?
028E 26 03 DEB NOPF1
0290 00 C027 S0PSEP J0R WITCH PASS OVER SEPARATOR
0293 79 0144 NOPF1 1ST RECSFL
0296 27 03 B00 NOTREX
0298 7E 0C07 JMP RECOVR PARSE SEPARATELY

```

```

029B 00 0336 NOTREX J0R GETS0V
029E 23 00 BCS SYNTA1
02A0 26 11 DEB J0NAME
02A2 DE 0143 L01 006701
02A5 07 03 STAA 3,1 SOURCE DRIVE
02A7 4F 04 CLD 0,1
02A9 20 13 ANBA WITURN

```

```

02AB 0F 40 SYNTA1 CLD PRIMARY
02AD DE 1000 L01 051000
02AF 7E 0AEE JMP ERRORS REPORT AND EXIT

```

```

02B3 DE 1143 J0NAME L01 006701 PROCESS SOURCE NAME
02B6 30 C029 J0R GETF1L
02B9 24 03 BCC NOTDRV
02BB 7E 0A39 EXOP03 JMP ERROR

```

```

02BE 00 76 WITURN B0R GETS0V
02C0 23 59 BCS SYNTA1
02C2 26 09 DEB J0NAME
02C4 DE 1203 L01 006701
02C7 07 03 STAA 3,1 DESTINATION DRIVE
02C9 4F 04 CLD 0,1
02CB 20 00 ANBA C00000

```

```

02CD DE 1203 B0NAME2 L01 006701 PROCESS DEST NAME
02D0 00 C029 J0R GETF1L
02D3 23 E6 BCS ERRORS

```

* CHECK FILE NAMES

```

*
* ENTER --- NAME/DRV IN FC00
* EXIT --- NUMBER OF SHIFL=0
* 1 NAME SHIFL=+1
* 2 NAMES SHIFL=+1
*

```

```

02D5 DE 114F CHILDAN L01 106701+12
02D8 06 03 L0AA 03
02DA A6 04 UPPER1 L0AA 0,1
02DC 01 40 C0PA 0040
02DE 25 04 BLD WITLO1
02E0 04 3F ANBA H0FF MAKE UPPER CASE
02E2 07 04 STAA 0,1
02E4 30 01 WITLO1 101
02E6 3A DEB DECO
02E7 26 F1 DEB UPPER1
02E9 DE 120F L01 106701+12
02EC 06 03 L0AA 03
02EE A6 04 UPPER2 L0AA 0,1
02F0 01 40 C0PA 0040
02F2 25 04 BLD WITLO2
02F4 04 3F ANBA H0FF MAKE UPPER CASE
02F6 07 04 STAA 0,1
02F8 30 01 WITLO2 101
02FA 3A DEB DECO
02FB 26 F1 DEB UPPER2
02FD 06 1206 L0AA DESFC0+3 MOVE DRIVES OVER
02FF 07 9E ANBA DES0V
0302 F6 1144 L0AA DESFC0+3
0305 07 90 STAA DES0V
0307 79 1147 TST DESFC0+4 SOURCE NAME?
0309 27 1C DEB ANAME JUST NUMBERS
030B 79 120F TST DESFC0+4
030D 27 14 DEB CHILDAN
030F 79 120F TST DESFC0+12 SECOND EXTENSION?
0311 26 0C DEB TH0000
0313 3E 114F L01 DESFC0+12 MOVE EXTENSION
0315 0F 120F STI DESFC0+12
0317 06 1151 L0AA DESFC0+14
0319 07 1291 STAA DESFC0+14
031B 04 47 ANBA DEC SHIFL FLAG TWO NAMES
031D 39 015 RTS

```

```

0320 DE A7 CHILDAN INC SHIFL FLAG ONE NAME
0322 39 015 RTS

0328 79 1207 W0NAME TST DESFC0+4 DEST GAVE WITHOUT
032B 26 06 DEB SYNTA3 SOURCE NAME?
032D 34 04 A1E0 C0R C0R TO DES0V
0331 26 27 DEB NOTAN
0333 7E 0240 SYNTA3 JMP SYNTA1 COULD BE SRC

```

```

* GET DRIVE FROM COMMAND LINE
* DETERMINE IF FILE NAME
*

```

```

0336 06 01 GETDRV L0AA 01
0338 DE CC14 L01 006701
033B 46 04 L0AA 0,1 READ BUFFER WITHOUT KEYIN
033D 00 C021 J0R CLARG
0340 25 10 BCS NOTAN
0342 01 39 C0PA 014
0344 22 13 B01 ALPAA
0346 06 01 L0AA 3,1
0348 01 2E C0PA 014 FILE NAME NEXT?
034A 27 00 DEB ALPAA
034C 04 03 ANBA 03
034E 3A 02 P0R 0
0350 00 C027 J0R WITCH
0353 00 C027 J0R WITCH
0356 33 02 P0L 0
0358 3F CLD0
0359 30 P0T0
035A 39 015 ALPAA RTS

```

```

*****

```

```

* READ TWO DIRECTORIES (0TH HEREIN)
* DELETED AND FILES IN SOURCE DIRECTORY
*

```

```

035D 06 01 READ20 L0AA 01 ANYTHING OR FOR DEST
035F 07 00 STAA 0000
0361 07 9F STAA 0000
0363 06 00 L01 DES0V0 START OF G000SPACE
0365 0F 0A B13 T0P0
0367 06 9E L0AA DES0V0
0369 00 10 B0R READ10
036B 0E 0A L01 T0P0

```



```

0360 9F 02      STI  SPCHIN  SAVE PRIOTED
0360 70 0140    TST  UPDTLG  UPDATE TO FILE DATE
0370 36 02      BNE  USEFLS  USEFLS
0372 0F 9F      CLR  BDATEX  DATEX
0374 0C A0      USEFL  BDATEX  BDATEX
0376 0C A0      INC  DIRFLD  FLAG SRC DIR
0378 96 90      LDM  SPCHIN  READ SOURCE DIR
037A 80 04      BSR  READ10  READ SOURCE DIR
037C 0F A0      CLR  DIRFLD
037E 9E 04      LBI  TEMP2
0380 9F 04      STI  BUFFER  SAVE PRIOTED
0382 12          NOP
0383 1C EF      CLI  CL1      ALLOW INTERRUPTS
0385 39          RETR3  RTS

```

* READ A DIRECTORY

- * LOADS 16 BYTES OF EACH ENTRY INTO
- * MEMORY, REJECTS BAD FILED THAT
- * HAVE ZERO FOR SIZE, TRACK, OR
- * SECTOR, REJECTS FILES WITH BAD
- * CREATION DATES.

```

0386 97 9C      READ10  BTAA  DRIVE
0388 0E C040    LBI  BTAA  OFC0
038A 07 03      BTAA  J,1
038C 06 05      LDM  BTAA  V5      FIRST DIR SECTOR
038E 07 00 1F  BTAA  91F,1
0390 0F 00 1E  CLR  91E,1
0392 0F 04      LDM  BTAA  99      READ SECTOR
0394 07 04      BTAA  0,1
0396 16 10      BLOOP 0E1      STOP (INTERRUPTS)
0398 0E C040    LBI  BTAA  OFC0
039A 00 0406    JSR  FIB
039C 27 03      BSR  RECDIR
039E 7E 01C0    JAP  RECDIR

```

```

03A6 0E C090    RECDIR  LBI  OFC0+050  START OF FIRST GAVE
03A8 9F 00      STI  TEMP1  FEB PRIOTED

```

* CHECK BAD FILE

```

03A8 A6 04      RECDIR  LDM  0,1      EMPTY GAVE?
03AA 27 04      BSR  RECDIR  END OF DIR?
03AC 2A 07      BPL  RECDIR  DELETED FILE?
03AE 70 0140    TST  RECDIR  COPY BY NUMBER?
03B0 27 20      BSR  SKIP2  IF NOT SKIP DELETED
03B2 20 72      BSR  GOODIR  IF NO KEEP DELETED

```

```

03B8 A6 00      NOTREL  LDM  13,1      DISK ADDRESS
03BA 27 17      BSR  BADFIL
03BC A6 0E      LDM  14,1
03BE 27 13      BSR  BADFIL
03C0 A6 0F      LDM  13,1
03C2 27 0F      BSR  BADFIL
03C4 A6 00 10   LDM  16,3
03C6 27 0A      BSR  BADFIL
03C8 60 00 12   TST  10,1      LBN FILE SIZE
03CA 26 1A      BNE  CHECKAT
03CC A6 00 11   TST  17,1      HBN FILE SIZE
03CE 26 15      BNE  CHECKAT
03D0 06 01      BADFIL  LDM  01
03D2 07 A1      BTAA  00010  NO COPY BY NUMBER
03D4 0E 0E0E    LBI  0000000
03D6 00 C01E    JSR  PSTIME
03D8 9E 00      LBI  TEMP1
03DA 00 00F4    JSR  OUTTAB
03DC 9E 00      LBI  TEMP1
03DE A0 40      TST  BADOK
03E0 27 73      SKIP2  BSR  SKIP  SKIP THIS FILE

```

* CHECK BAD DATE

```

03E8 A6 00 15   CHECKAT  LDM  21,1      MONTH
03EA 27 10      BSR  BADATE
03EC 01 0C      CMPA  012
03EE 22 14      BNE  BADATE
03F0 A6 00 16   LDM  22,1      DAY
03F2 27 0F      BSR  BADATE
03F4 06 01      CMPA  011
03F6 22 09      BNE  BADATE
03F8 A6 00 17   LDM  23,1      YEAR
03FA 27 0C      CMPA  076
03FC 23 04      BNE  BADATE
0400 01 04      CMPA  01000
0402 23 25      BSR  000010  DIR ENTRY ACCEPTED
0404 06 01      BADATE  LDM  01
0406 07 A1      BTAA  00010  NO COPY BY NUMBER
0408 00 9F      TST  BATERK
040A 2E 10      BGT  00010  IF NOT BATERK DISPLAY
040C 0E 0E00    LBI  0007000

```

```

0410 00 C01E    JSR  PSTIME
0412 9E 00      LBI  TEMP1
0414 30 00F4    JSR  OUTTAB  BAD DATE FILE
0416 00 9F      TST  BATERK
0418 20 3F      BNE  SKIP  ONLY PROMPT ONCE
041C 0E 0EAD    LBI  0000000
041E 00 01C0    JSR  ASKDIR
0420 27 04      BSR  GOODIR
0422 06 9F      DEC  BATERK  FLAG REJECT BAD DATES
0424 20 33      BSR  SKIP
0426 0C 9F      GOODIR  INC  BATERK  FLAG BAD DATES ALLOWED

```

* FILE JUMBED SK, ADD TO RAM

```

0426 9E 00      GOODIR  LBI  TEMP1
0428 CA 00      LDM  011      SAME PLUS ENTER
042E 80 1A      MOVNAM  BSR  MOVNAM
0430 5A          DEC3
0432 26 F0      BNE  MOVNAM
0434 CA 06      LDM  04
0436 30 91      MOVOME  IN3      SKIP ATTRIB, RESERVED, ADDR
0438 5A          DEC3
043A 26 F0      BNE  MOVOME
043C 00 0E      BSR  MOVNAM  FILE SIZE
043E 30 01      BSR  MOVNAM
0440 30 01      BSR  MOVNAM  SECTOR MAP
0442 00 06      BSR  MOVNAM  RESERVED
0444 00 04      BSR  MOVNAM  CREATION DATE
0446 00 02      BSR  MOVNAM
0448 20 16      BSR  MOVNAM

```

```

044A A6 04      MOVNAM  LDM  0,1
044C 30 01      BSR  IN3
044E 9F 00      STI  TEMP1
0450 9E 0A      LBI  TEMP2
0452 A7 04      BTAA  0,1      MOVE OVER
0454 30 01      BSR  IN3
0456 9F 0A      STI  TEMP2
0458 9E 00      LBI  TEMP1
045A 39          RETR3  RTS

```

* ADVANCE TO NEXT FILE AND CHECK END

```

045B C6 10      SKIP  LDM  024      SIZE OF ENTRY
045D 00 00AC    CHECKED  JSR  MOVNAM
045F 96 0A      LDM  024      TEMP2
0461 91 06      CMPA  00000  OUT OF SPACE?
0463 27 13      BNE  OVERLO
0465 0C C700    CP1  OFC0+320  NONE?
0467 27 03      BSR  BADLINE
0469 7E 03A0    JAP  MOVNAM
046B 00 0000    MOV,INC  LBI  FC0+044  LINE
046D 0F C00E    STI  FC0+04E
046F 27 E4      BSR  RETNAM  EXTEND-ED OF DIR
0471 7E 0399    JAP  BLOOP

```

```

0479 0E 0700    OVERLO  LBI  MOVNAM  DIRECTORIES WON'T FIT
047C 0F A0      CLR  PUTNAM
047E 7E 0AEE    JAP  EXHIB

```

* DISPLAY HELP MESSAGE

```

0481 96 A4      HELP  LDM  PAUSE
0483 97 EC09    LDM  07AA  PAUSE
0485 96 A3      LDM  OUTTAB  RESTORE OLD PAUSE
0487 07 CC22    STAA  OUTTAB  RESTORE OLD OUT SWITCH
0489 0E CC33    LBI  0ALPHE1  GENERAL
048B 00 C01E    JSR  PSTIME
048D 0E CC73    LBI  0ALPHE2  FILE NUMBER
048F 00 C01E    JSR  PSTIME
0491 0E CC93    LBI  0ALPHE3  SECTOR RECOVER
0493 00 C01E    JSR  PSTIME
0495 00 C024    JSR  PCRLF
0497 0E 013C    LBI  0ALPFLB
0499 9F 00      STI  11001  OPTION FLAG PRIOTED
04A1 0E 0C03    LBI  00P7000
04A3 9F 00      STI  TEMP2  SAVE MESSAGE
04A5 9E 00      LBI  TEMP1
04A7 A6 04      LDM  0,1      NET FLAG
04A9 30 01      BSR  IN3
04AB 9F 00      STI  TEMP1
04AD 0E 0E00    LBI  0ALPHE4
04AF 4B 03      TSTA  1000
04B1 27 03      BSR  1000  SWITCH TO YES
04B3 0E 0E07    LBI  0ALPHE5  OUTPUT YES OR NO
04B5 00 C01E    JSR  PSTIME

```

```

040E 0E 0A      LBI TEMP2  RESUME MESSAGE
040D 00 0813    JBR PMATA
040C 30 01      JBI      BRIP ETS
040B 0C 0E0F    CFI 0LSTOP1+1
040A 26 0E      BNE HLPLOP
0409 7E 0A0F    JMP EXIT

```

* OUTPUT MESSAGE AND GET RESPONSE

```

*
* ENTER --- MESSAGE
* EXIT --- EXIT DUMPED CASE
*

```

```

040D 00 C01E    ASKMSG JBR PSTMSG
040D 00 C013    JBR BETCHN
0403 01 00      CFI 0A0D  LOWER CASE
0405 25 02      BLO NOTLO
0407 04 3F      BNE 040F
0409 01 3F      BNE 040F
040B 19 00      RETR2 RTS

```

* SOURCE DIRECTORY SORT

```

*
* ENTER --- SOURCE DIRECTORY LOADED
*

```

```

040C 0E 02      SORT LBI SRCDIR
040E 0F 00      STI TEMP1  END OF SCAN
040D 0A 04      LDA BUTFR  BACK UP TWO ENTRIES
040E 0A 03      LDA BUTFR+1
040A 09 10      BEND 01A
040A 02 00      SRCA 00
040B 01 00      CFI 0E01  ONLY ONE ENTRY?
040A 26 04      BNE NOTONE
040C 01 0F      CFI 0E01+1
040E 27 00      BNE RETR2
040D 09 10      BEND 01A
0407 02 00      SRCA 00
0404 07 0C      STAB TEMP3  NEXT TO LAST ITEM
040A 07 00      STAB TEMP3+1
040B 20 00      BNA SORT2

```

* MAIN LOOP SETS UP POINTERS

```

040A 00 A2      SORT2 TBT SUPFLD  NO SWAPS LAST PASS?
040C 26 00      BNE RETR2
040E 0E 00      LBI TEMP1  MOVING TOP END
040D 0C 0C      CFI 0E01  FIXED BOTTOM END
040E 27 07      BNE RETR2  DONE WHEN MEET
040A 00 0A0A    JBR ADD16  SWAP TEMP1
0407 0E 0C      SORT2 LBI TEMP3  NEXT TO LAST
0409 0F 0A      STI TEMP2  CURRENT ADDRESS
040B 0A 01      LDA BI  SET FLAG-NO SWAP
040B 07 A2      STAB SUPFLD

```

* COMPARE TWO ITEMS

```

0409 0A 00      COMPARE LDAI 011  COMPARE SIZE
040E 0A 04      COMPARE LDAI 01A  GET FIRST CHAR
040B 01 00 10   CFI 0E01  SECOND CHAR
040A 25 07      BCS INCHRN
040B 26 10      BNE SWAP
040A 30 01      JBI
040C 30 00      BEND
040B 26 F2      BNE COMPARE
040F 0E 0A      LBI TEMP1  PAGE OVER?
040E 0E 00      CFI 0E01
040C 27 05      BCS 040F
040E 26 0A      LDA TEMP2+1
0407 0A 00      BEND 01A  MOVE FORWARD TEMP1
040B 00 00      STAB TEMP2+1
040B 02 00      SRCA 00
040E 07 0A      STAB TEMP2
040B 0E 0A      LBI TEMP2
040C 20 0A      BNA COMPARE

```

* SWAP TWO ITEMS

* STRAIGHT LINE CODE WHEN FOR SPEED

```

040B 0E 0A      SWAP LBI TEMP2  START OF ITER
040C 0A 04      LDAI 01A
040B 0E 10      LDAI 1A,1
040C 07 10      STAB 1A,2
040E 07 04      STAB 01A
040A 0A 01      LDAI 1A,1
040C 0A 0E 11   LDAI 17,1

```

```

040A 07 00 11   STAB 17,1
0409 07 01      STAB 1,1
040A 0A 02      LDAI 2,2
040B 0A 10      LDAI 10,1
040A 07 00 12   STAB 10,1
040E 07 02      STAB 2,1
040E 0A 03      LDAI 3,3
040E 0A 00 13   LDAI 13,1
040A 07 00 13   STAB 13,1
040E 07 03      STAB 3,1
040E 0A 04      LDAI 4,4
040A 0E 14      LDAI 14,1
040A 07 00 14   STAB 14,1
040E 07 04      STAB 4,1
040E 0A 05      LDAI 5,5
040A 0E 15      LDAI 15,1
040E 07 05      STAB 5,1
040E 0A 06      LDAI 6,6
040E 0A 00 16   LDAI 16,1
040E 07 06      STAB 6,6
040E 0A 07      LDAI 7,7
040E 0A 00 17   LDAI 17,1
040E 07 07      STAB 7,7
040E 0A 08      LDAI 8,8
040E 0A 00 18   LDAI 18,1
040E 07 08      STAB 8,8
040E 0A 09      LDAI 9,9
040E 0A 00 19   LDAI 19,1
040E 07 09      STAB 9,9
040E 0A 0A      LDAI 10,10
040E 0A 00 1A   LDAI 1A,1
040E 07 0A      STAB 1A,1
040E 0A 0B      LDAI 11,11
040E 0A 00 1B   LDAI 1B,1
040E 07 0B      STAB 1B,1
040E 0A 0C      LDAI 12,12
040E 0A 00 1C   LDAI 1C,1
040E 07 0C      STAB 1C,1
040E 0A 0D      LDAI 13,13
040E 0A 00 1D   LDAI 1D,1
040E 07 0D      STAB 1D,1
040E 0A 0E      LDAI 14,14
040E 0A 00 1E   LDAI 1E,1
040E 07 0E      STAB 1E,1
040E 0A 0F      LDAI 15,15
040E 0A 00 1F   LDAI 1F,1
040E 07 0F      STAB 1F,1
040E 0A 10      LDAI 16,16
040E 0A 00 20   LDAI 20,1
040E 07 10      STAB 20,1
040E 0A 11      LDAI 17,17
040E 0A 00 21   LDAI 21,1
040E 07 11      STAB 21,1
040E 0A 12      LDAI 18,18
040E 0A 00 22   LDAI 22,1
040E 07 12      STAB 22,1
040E 0A 13      LDAI 19,19
040E 0A 00 23   LDAI 23,1
040E 07 13      STAB 23,1
040E 0A 14      LDAI 20,20
040E 0A 00 24   LDAI 24,1
040E 07 14      STAB 24,1
040E 0A 15      LDAI 21,21
040E 0A 00 25   LDAI 25,1
040E 07 15      STAB 25,1
040E 0A 16      LDAI 22,22
040E 0A 00 26   LDAI 26,1
040E 07 16      STAB 26,1
040E 0A 17      LDAI 23,23
040E 0A 00 27   LDAI 27,1
040E 07 17      STAB 27,1
040E 0A 18      LDAI 24,24
040E 0A 00 28   LDAI 28,1
040E 07 18      STAB 28,1
040E 0A 19      LDAI 25,25
040E 0A 00 29   LDAI 29,1
040E 07 19      STAB 29,1
040E 0A 20      LDAI 26,26
040E 0A 00 30   LDAI 30,1
040E 07 20      STAB 30,1
040E 0A 21      LDAI 27,27
040E 0A 00 31   LDAI 31,1
040E 07 21      STAB 31,1
040E 0A 22      LDAI 28,28
040E 0A 00 32   LDAI 32,1
040E 07 22      STAB 32,1
040E 0A 23      LDAI 29,29
040E 0A 00 33   LDAI 33,1
040E 07 23      STAB 33,1
040E 0A 24      LDAI 30,30
040E 0A 00 34   LDAI 34,1
040E 07 24      STAB 34,1
040E 0A 25      LDAI 31,31
040E 0A 00 35   LDAI 35,1
040E 07 25      STAB 35,1
040E 0A 26      LDAI 32,32
040E 0A 00 36   LDAI 36,1
040E 07 26      STAB 36,1
040E 0A 27      LDAI 33,33
040E 0A 00 37   LDAI 37,1
040E 07 27      STAB 37,1
040E 0A 28      LDAI 34,34
040E 0A 00 38   LDAI 38,1
040E 07 28      STAB 38,1
040E 0A 29      LDAI 35,35
040E 0A 00 39   LDAI 39,1
040E 07 29      STAB 39,1
040E 0A 30      LDAI 36,36
040E 0A 00 40   LDAI 40,1
040E 07 30      STAB 40,1
040E 0A 31      LDAI 37,37
040E 0A 00 41   LDAI 41,1
040E 07 31      STAB 41,1
040E 0A 32      LDAI 38,38
040E 0A 00 42   LDAI 42,1
040E 07 32      STAB 42,1
040E 0A 33      LDAI 39,39
040E 0A 00 43   LDAI 43,1
040E 07 33      STAB 43,1
040E 0A 34      LDAI 40,40
040E 0A 00 44   LDAI 44,1
040E 07 34      STAB 44,1
040E 0A 35      LDAI 41,41
040E 0A 00 45   LDAI 45,1
040E 07 35      STAB 45,1
040E 0A 36      LDAI 42,42
040E 0A 00 46   LDAI 46,1
040E 07 36      STAB 46,1
040E 0A 37      LDAI 43,43
040E 0A 00 47   LDAI 47,1
040E 07 37      STAB 47,1
040E 0A 38      LDAI 44,44
040E 0A 00 48   LDAI 48,1
040E 07 38      STAB 48,1
040E 0A 39      LDAI 45,45
040E 0A 00 49   LDAI 49,1
040E 07 39      STAB 49,1
040E 0A 40      LDAI 46,46
040E 0A 00 50   LDAI 50,1
040E 07 40      STAB 50,1
040E 0A 41      LDAI 47,47
040E 0A 00 51   LDAI 51,1
040E 07 41      STAB 51,1
040E 0A 42      LDAI 48,48
040E 0A 00 52   LDAI 52,1
040E 07 42      STAB 52,1
040E 0A 43      LDAI 49,49
040E 0A 00 53   LDAI 53,1
040E 07 43      STAB 53,1
040E 0A 44      LDAI 50,50
040E 0A 00 54   LDAI 54,1
040E 07 44      STAB 54,1
040E 0A 45      LDAI 51,51
040E 0A 00 55   LDAI 55,1
040E 07 45      STAB 55,1
040E 0A 46      LDAI 52,52
040E 0A 00 56   LDAI 56,1
040E 07 46      STAB 56,1
040E 0A 47      LDAI 53,53
040E 0A 00 57   LDAI 57,1
040E 07 47      STAB 57,1
040E 0A 48      LDAI 54,54
040E 0A 00 58   LDAI 58,1
040E 07 48      STAB 58,1
040E 0A 49      LDAI 55,55
040E 0A 00 59   LDAI 59,1
040E 07 49      STAB 59,1
040E 0A 50      LDAI 56,56
040E 0A 00 60   LDAI 60,1
040E 07 50      STAB 60,1
040E 0A 51      LDAI 57,57
040E 0A 00 61   LDAI 61,1
040E 07 51      STAB 61,1
040E 0A 52      LDAI 58,58
040E 0A 00 62   LDAI 62,1
040E 07 52      STAB 62,1
040E 0A 53      LDAI 59,59
040E 0A 00 63   LDAI 63,1
040E 07 53      STAB 63,1
040E 0A 54      LDAI 60,60
040E 0A 00 64   LDAI 64,1
040E 07 54      STAB 64,1
040E 0A 55      LDAI 61,61
040E 0A 00 65   LDAI 65,1
040E 07 55      STAB 65,1
040E 0A 56      LDAI 62,62
040E 0A 00 66   LDAI 66,1
040E 07 56      STAB 66,1
040E 0A 57      LDAI 63,63
040E 0A 00 67   LDAI 67,1
040E 07 57      STAB 67,1
040E 0A 58      LDAI 64,64
040E 0A 00 68   LDAI 68,1
040E 07 58      STAB 68,1
040E 0A 59      LDAI 65,65
040E 0A 00 69   LDAI 69,1
040E 07 59      STAB 69,1
040E 0A 60      LDAI 66,66
040E 0A 00 70   LDAI 70,1
040E 07 60      STAB 70,1
040E 0A 61      LDAI 67,67
040E 0A 00 71   LDAI 71,1
040E 07 61      STAB 71,1
040E 0A 62      LDAI 68,68
040E 0A 00 72   LDAI 72,1
040E 07 62      STAB 72,1
040E 0A 63      LDAI 69,69
040E 0A 00 73   LDAI 73,1
040E 07 63      STAB 73,1
040E 0A 64      LDAI 70,70
040E 0A 00 74   LDAI 74,1
040E 07 64      STAB 74,1
040E 0A 65      LDAI 71,71
040E 0A 00 75   LDAI 75,1
040E 07 65      STAB 75,1
040E 0A 66      LDAI 72,72
040E 0A 00 76   LDAI 76,1
040E 07 66      STAB 76,1
040E 0A 67      LDAI 73,73
040E 0A 00 77   LDAI 77,1
040E 07 67      STAB 77,1
040E 0A 68      LDAI 74,74
040E 0A 00 78   LDAI 78,1
040E 07 68      STAB 78,1
040E 0A 69      LDAI 75,75
040E 0A 00 79   LDAI 79,1
040E 07 69      STAB 79,1
040E 0A 70      LDAI 76,76
040E 0A 00 80   LDAI 80,1
040E 07 70      STAB 80,1
040E 0A 71      LDAI 77,77
040E 0A 00 81   LDAI 81,1
040E 07 71      STAB 81,1
040E 0A 72      LDAI 78,78
040E 0A 00 82   LDAI 82,1
040E 07 72      STAB 82,1
040E 0A 73      LDAI 79,79
040E 0A 00 83   LDAI 83,1
040E 07 73      STAB 83,1
040E 0A 74      LDAI 80,80
040E 0A 00 84   LDAI 84,1
040E 07 74      STAB 84,1
040E 0A 75      LDAI 81,81
040E 0A 00 85   LDAI 85,1
040E 07 75      STAB 85,1
040E 0A 76      LDAI 82,82
040E 0A 00 86   LDAI 86,1
040E 07 76      STAB 86,1
040E 0A 77      LDAI 83,83
040E 0A 00 87   LDAI 87,1
040E 07 77      STAB 87,1
040E 0A 78      LDAI 84,84
040E 0A 00 88   LDAI 88,1
040E 07 78      STAB 88,1
040E 0A 79      LDAI 85,85
040E 0A 00 89   LDAI 89,1
040E 07 79      STAB 89,1
040E 0A 80      LDAI 86,86
040E 0A 00 90   LDAI 90,1
040E 07 80      STAB 90,1
040E 0A 81      LDAI 87,87
040E 0A 00 91   LDAI 91,1
040E 07 81      STAB 91,1
040E 0A 82      LDAI 88,88
040E 0A 00 92   LDAI 92,1
040E 07 82      STAB 92,1
040E 0A 83      LDAI 89,89
040E 0A 00 93   LDAI 93,1
040E 07 83      STAB 93,1
040E 0A 84      LDAI 90,90
040E 0A 00 94   LDAI 94,1
040E 07 84      STAB 94,1
040E 0A 85      LDAI 91,91
040E 0A 00 95   LDAI 95,1
040E 07 85      STAB 95,1
040E 0A 86      LDAI 92,92
040E 0A 00 96   LDAI 96,1
040E 07 86      STAB 96,1
040E 0A 87      LDAI 93,93
040E 0A 00 97   LDAI 97,1
040E 07 87      STAB 97,1
040E 0A 88      LDAI 94,94
040E 0A 00 98   LDAI 98,1
040E 07 88      STAB 98,1
040E 0A 89      LDAI 95,95
040E 0A 00 99   LDAI 99,1
040E 07 89      STAB 99,1
040E 0A 90      LDAI 96,96
040E 0A 00 100  LDAI 100,1
040E 07 90      STAB 100,1
040E 0A 91      LDAI 97,97
040E 0A 00 101  LDAI 101,1
040E 07 91      STAB 101,1
040E 0A 92      LDAI 98,98
040E 0A 00 102  LDAI 102,1
040E 07 92      STAB 102,1
040E 0A 93      LDAI 99,99
040E 0A 00 103  LDAI 103,1
040E 07 93      STAB 103,1
040E 0A 94      LDAI 100,100
040E 0A 00 104  LDAI 104,1
040E 07 94      STAB 104,1
040E 0A 95      LDAI 101,101
040E 0A 00 105  LDAI 105,1
040E 07 95      STAB 105,1
040E 0A 96      LDAI 102,102
040E 0A 00 106  LDAI 106,1
040E 07 96      STAB 106,1
040E 0A 97      LDAI 103,103
040E 0A 00 107  LDAI 107,1
040E 07 97      STAB 107,1
040E 0A 98      LDAI 104,104
040E 0A 00 108  LDAI 108,1
040E 07 98      STAB 108,1
040E 0A 99      LDAI 105,105
040E 0A 00 109  LDAI 109,1
040E 07 99      STAB 109,1
040E 0A 100     LDAI 106,106
040E 0A 00 110  LDAI 110,1
040E 07 100     STAB 110,1
040E 0A 101     LDAI 107,107
040E 0A 00 111  LDAI 111,1
040E 07 101     STAB 111,1
040E 0A 102     LDAI 108,108
040E 0A 00 112  LDAI 112,1
040E 07 102     STAB 112,1
040E 0A 103     LDAI 109,109
040E 0A 00 113  LDAI 113,1
040E 07 103     STAB 113,1
040E 0A 104     LDAI 110,110
040E 0A 00 114  LDAI 114,1
040E 07 104     STAB 114,1
040E 0A 105     LDAI 111,111
040E 0A 00 115  LDAI 115,1
040E 07 105     STAB 115,1
040E 0A 106     LDAI 112,112
040E 0A 00 116  LDAI 116,1
040E 07 106     STAB 116,1
040E 0A 107     LDAI 113,113
040E 0A 00 117  LDAI 117,1
040E 07 107     STAB 117,1
040E 0A 108     LDAI 114,114
040E 0A 00 118  LDAI 118,1
040E 07 108     STAB 118,1
040E 0A 109     LDAI 115,115
040E 0A 00 119  LDAI 119,1
040E 07 109     STAB 119,1
040E 0A 110     LDAI 116,116
040E 0A 00 120  LDAI 120,1
040E 07 110     STAB 120,1
040E 0A 111     LDAI 117,117
040E 0A 00 121  LDAI 121,1
040E 07 111     STAB 121,1
040E 0A 112     LDAI 118,118
040E 0A 00 122  LDAI 122,1
040E 07 112     STAB 122,1
040E 0A 113     LDAI 119,119
040E 0A 00 123  LDAI 123,1
040E 07 113     STAB 123,1
040E 0A 114     LDAI 120,120
040E 0A 00 124  LDAI 124,1
040E 07 114     STAB 124,1
040E 0A 115     LDAI 121,121
040E 0A 00 125  LDAI 125,1
040E 07 115     STAB 125,1
040E 0A 116     LDAI 122,122
040E 0A 00 126  LDAI 126,1
040E 07 116     STAB 126,1
040E 0A 117     LDAI 123,123
040E 0A 00 127  LDAI 127,1
040E 07 117     STAB 127,1
040E 0A 118     LDAI 124,124
040E 0A 00 128  LDAI 128,1
040E 07 118     STAB 128,1
040E 0A 119     LDAI 125,125
040E 0A 00 129  LDAI 129,1
040E 07 119     STAB 129,1
040E 0A 120     LDAI 126,126
040E 0A 00 130  LDAI 130,1
040E 07 120     STAB 130,1
040E 0A 121     LDAI 127,127
040E 0A 00 131  LDAI 131,1
040E 07 121     STAB 131,1
040E 0A 122     LDAI 128,128
040E 0A 00 132  LDAI 132,1
040E 07 122     STAB 132,1
040E 0A 123     LDAI 129,129
040E 0A 00 133  LDAI 133,1
040E 07 123     STAB 133,1
040E 0A 124     LDAI 130,130
040E 0A 00 134  LDAI 134,1
040E 07 124     STAB 134,1
040E 0A 125     LDAI 131,131
040E 0A 00 135  LDAI 135,1
040E 07 125     STAB 135,1
040E 0A 126     LDAI 132,132
040E 0A 00 136  LDAI 136,1
040E 07 126     STAB 136,1
040E 0A 127     LDAI 133,133
040E 0A 00 137  LDAI 137,1
040E 07 127     STAB 137,1
040E 0A 128     LDAI 134,134
040E 0A 00 138  LDAI 138,1
040E 07 128     STAB 138,1
040E 0A 129     LDAI 135,135
040E 0A 00 139  LDAI 139,1
040E 07 129     STAB 139,1
040E 0A 130     LDAI 136,136
040E 0A 00 140  LDAI 140,1
040E 07 130     STAB 140,1
040E 0A 131     LDAI 137,137
040E 0A 00 141  LDAI 141,1
040E 07 131     STAB 141,1
040E 0A 132     LDAI 138,138
040E 0A 00 142  LDAI 142,1
040E 07 132     STAB 142,1
040E 0A 133     LDAI 139,139
040E 0A 00 143  LDAI 143,1
040E 07 133     STAB 143,1
040E 0A 134     LDAI 140,140
040E 0A 00 144  LDAI 144,1
040E 07 134     STAB 144,1
040E 0A 135     LDAI 141,141
040E 0A 00 145  LDAI 145,1
040E 07 135     STAB 145,1
040E 0A 136     LDAI 142,142
040E 0A 00 146  LDAI 146,1
040E 07 136     STAB 146,1
040E 0A 137     LDAI 143,143
040E 0A 00 147  LDAI 147,1
040E 07 137     STAB 147,1
040E 0A 138     LDAI 144,144
040E 0A 00 148  LDAI 148,1
040E 07 138     STAB 148,1
040E 0A 139     LDAI 145,145
040E 0A 00 149  LDAI 149,1
040E 07 139     STAB 149,1
040E 0A 140     LDAI 146,146
040E 0A 00 150  LDAI 150,1
040E 07 140     STAB 150,1
040E 0A 141     LDAI 147,147
040E 0A 00 151  LDAI 151,1
040E 07 141     STAB 151,1
040E 0A 142     LDAI 148,148
040E 0A 00 152  LDAI 152,1
040E 07 142     STAB 152,1
040E 0A 143     LDAI 149,149
040E 0A 00 153  LDAI 153,1
040E 07 143     STAB 153,1
040E 0A 144     LDAI 150,150
040E 0A 00 154  LDAI 154,1
040E 07 144     STAB 154,1
040E 0A 145     LDAI 151,151
040E 0A 00 155  LDAI 155,1
040E 07 145     STAB 155,1
040E 0A 146     LDAI 152,152
040E 0A 00 156  LDAI 156,1
040E 07 146     STAB 156,1
040E 0A 147     LDAI 153,153
040E 0A 00 157  LDAI 157,1
040E 07 147     STAB 157,1
040E 0A 148     LDAI 154,154
040E 0A 00 158  LDAI 158,1
040E 07 148     STAB 158,1
040E 0A 149     LDAI 155,155
040E 0A 00 159  LDAI 159,1
040E 07 149     STAB 159,1
040E 0A 150     LDAI 156,156
040E 0A 00 160  LDAI 160,1
040E 07 150     STAB 160,1
040E 0A 151     LDAI 157,157
040E 0A 00 161  LDAI 161,1
040E 07 151     STAB 161,1
040E 0A 152     LDAI 158,158
040E 0A 00 162  LDAI 162,1
040E 07 152     STAB 162,1
040E 0A 153     LDAI 159,159
040E 0A 00 163  LDAI 163,1
040E 07 153     STAB 163,1
040E 0A 154     LDAI 160,160
040E 0A 00 164  LDAI 164,1
040E 07 154     STAB 164,1
040E 0A 155     LDAI 161,161
040E 0A 00 165  LDAI 165,1
040E 07 155     STAB 165,1
040E 0A 156     LDAI 162,162
040E 0A 00 166  LDAI 166,1
040E 07 156     STAB 166,1
040E 0A 157     LDAI 163,163
040E 0A 00 167  LDAI 167,1
040E 07 157     STAB 167,1
040E 0A 158     LDAI 164,164
040E 0A 00 168  LDAI 168,1
040E 07 158     STAB 168,1
040E 0A 159     LDAI 165,165
040E 0A 00 169  LDAI 169,1
040E 07 159    
```

```

0400 0E 0A      LBI TEMP2
0400 0E 0B      CMA 13,1  BEST MONTH
0400 22 13      BUI KEEP
0410 25 0A      BUI KILL
0413 0E 0B      LBI TEMP1
0415 0A 0E      LMA 14,1  SOURCE DAY
0417 0E 0A      LBI TEMP2
0419 0E 0E      CMA 14,1  BEST DAY
0420 22 07      BUI KEEP
0420 70 0141    KILL PST KILLFLG REVERSE ACTION?
0420 27 07      BUI KILL2
0422 20 09      BUI KEEP2

0424 70 0141    KEEP TST KILLFLG REVERSE ACTION?
0427 27 04      BUI KEEP2
0429 0E 0B      KILL2 LBI TEMP1 SOURCE FILE
0429 0F 04      CLJ 0,1  FLAG NOT REVERSED
0429 0F 0404    KEEP2 JSR A0016  FLAG NOT REVERSED
0430 0C 04      CPE 00000000  DONE?
0432 26 06      BUI SELECT
0434 39      RE:MM RTS

```

* COPY SINGLE FILE

```

0435 70 0140    SINGLE TST FILEFLG COPY BY NAME?
0435 27 0B      BUI SINGLE
0435 0F 0A      CLJ 0,1  PATTERN
0435 0E 0E0C    LBI 00000000  NO COPY ALLOWED
0437 2E 0A0C    JRP 00000000

0442 00 07      SINGLE PST SINGLEL ONE OF TWO NAMES
0444 20 0B      BUI TROFIL
0446 0E 1147    LBI 00000000+4 REMOVE ONE NAME
0449 0F 0B      STE TEMP1
0449 00 07E6    JSR 00000000+4
044E 0E 1147    TROFIL LBI 00000000+4
0451 0F 00      STE TEMP2
0453 0E 02      LBI SRC010
0453 00 00C0    JSR F0011  FIND NAME IN SRC010
0458 24 06      BUI 00000000
0458 0E 0040    LBI 0000
0458 04 04      LMA 04  FILE NOT FOUND
0459 07 01      STAM 1,1
0461 7E 0439    JRP 00000000

```

```

0464 0E 0A      00000000 LBI TEMP2
0464 0F 0B      STE TEMP1 SRC NAME IN SRC DIR
0464 0F 07      CLJ 0,1  USED BY FIND FILE
0464 00 00E7    JSR C0011  NO COPY
0466 7E 0A0F    JRP 00000000

```

* COMPARE FILES TO MATCH LIST

* PARSE COMMAND LINE FOR MATCH STRING

```

0470 00 0024    MATCH JSR PC015
0473 00 0024    JRP PC015
0476 0A 0E11    LMA LST100
0479 01 0B      CMA 000
0479 27 03      BUI 00000000  NO MATCH STRING?
0479 31 0002    CMA 00000000
0480 26 04      BUI 00000000
0482 27 0A      00000000 STAM 00000000 SET NO STRING?
0484 20 03      BUI 00000000
0484 0E 0000    MATCH LBI 00000000+4 CLEAR NAME SPACE
0489 0F 04      MATCH CLJ 0,1
0489 30 01      IMJ 1
0489 0C 00C0    CPJ 0,1  BELT+4
0490 26 07      BUI MATCHA
0492 0E 0E14    LBI 00000000
0495 0A 04      LMA 0,1  GET MATCH CHAR
0497 00 0021    JSR CLASS
0498 24 10      BUI 00000000+4
049C 01 2E      CMA 0,1  EXTENSION?
049E 27 16      BUI PER100
04A0 01 00      CMA 000
04A2 27 00      BUI 00000000+4
04A4 01 0002    CMA 00000000+4
04A7 27 0B      BUI 00000000+4
04A9 7E 02A0    STMT12 JRP 00000000+4 ILLLEGAL CHAR

```

```

04AC 01 39      FILETB CMA 0,1
04AE 23 09      BUI STMT12 00000000+4
04B0 0E 0000    LBI 00000000+4
04B3 00 0020    JSR 00000000+4
04B4 25 01      BUI STMT12
04B7 20 21      BUI MATCHB

```

```

04BA 0E 0E14    PER100 LBI 00000000+4
04BB 30 01      IMJ 1
04BB 0F 0E14    STE 00000000+4
04C2 0A 04      LMA 04  EXTENSION SPACE
04C4 0E 000C    PER102 LBI 00000000+4
04C7 00 00C7    JSR 00000000+4
04CA 25 0F      BUI 00000000+4
04CC 01 00      CMA 00000000+4
04CE 25 02      BUI 00000000+4
04D0 04 5F      LMA 04 0,1  NAME UPPER CASE
04D2 07 04      MOTLOW STAM 0,1
04D4 30 01      IMJ 1
04D6 30 01      DEC 1
04D7 26 0E      BUI PER102
04D9 20 0E      BUI STMT12 00000000+4

```

* MATCH STRING IN NAME SPACE
* SEARCHED SOURCE DIR FOR MATCH

```

04DB 0E 02      MATCHES LBI SRC010
04DB 0F 0B      STE TEMP1
04DB 0F 0C      SERLOP STJ TEMP3
04E1 00 04      TST 0,1
04E3 27 0B      BUI 00000000+4  ALREADY FLAGGED
04E5 00 0A      TST 00000000+4
04E7 26 20      BUI 00000000+4  IF NO MATCH STRING
04E9 0E 0004    LBI 00000000+4
04EC 0F 0B      STE TEMP2
04EE 0A 0B      LMA 0B  NAME+EXT
04F0 0E 04      SERLO2 LBI TEMP3
04F2 0A 0C      LMA 0,1  GET DIR CHAR
04F4 30 01      IMJ 1
04F6 0F 0C      STE TEMP3
04F8 0E 0A      LBI TEMP2
04FA 00 04      TST 0,1  ZERO-DON'T CARE
04FC 27 04      BUI 00000000+4
04FE 01 04      CMA 0,1  MATCH?
0500 26 0E      BUI 00000000+4
0502 30 01      IMJ 1
0504 0F 0B      STJ TEMP2
0506 3A 01      DEC 1
0507 26 07      BUI SERLO2

```

* FOUND MATCH OR BAD MATCH

```

0509 00 0005    F0011 JSR 00000000+4 REPORT
0509 70 0141    TST 00000000+4
050F 26 10      BUI 00000000+4
0511 70 0145    TST 00000000+4
0514 27 10      BUI 00000000+4
0516 0E 0075    LBI 00000000+4
0519 00 00C0    JSR 00000000+4
051C 27 10      BUI 00000000+4
051E 01 50      CMA 0,1
0520 26 09      BUI 00000000+4
0522 7F 0145    CLJ 00000000+4
0525 20 07      BUI 00000000+4

```

```

0527 01 44      MOTPHY CMA 0,1
0529 26 0E      BUI 00000000+4
0529 7E 0A0F    E0114 JRP 00000000+4
052E 70 0117    TRYEST TST 00000000+4
0531 26 06      BUI 00000000+4
0533 00 07E1    JSR 00000000+4
0536 00 00E7    JSR 00000000+4
0539 00 0C      SERLOP TST 00000000+4
0539 27 0F      BUI 00000000+4
0539 0E 00      LBI 00000000+4
053F 0A 03      LMA 00000000+4
0541 07 0E22    STAM 00000000+4
0544 00 0024    JSR PC015
0547 0A 0F      LMA 00000000+4
0549 07 0E22    STAM 00000000+4
054C 0E 0B      00000000+4 LBI TEMP1
054E 0F 04      CLJ 0,1  DELETE AFTER COPY
0550 00 0040    B0011 JSR 00000000+4
0553 0E 04      CPE 00000000+4
0553 26 0B      BUI 00000000+4
0557 7E 0404    JRP 00000000+4

```

* COPY BY DIRECTORY NAME

```

055A 00 41      00000000+4 LBI 00000000+4
055C 27 03      BUI 00000000+4
055E 0E 0003    LBI 00000000+4
0561 20 10      BUI 00000000+4

```



```

0763 OF 91      MVSAD CLR NIBAND FLAG NO RANGE
0763 OF 92      CLR NIBAND+1
0763 BA CC11    LDA LSTTAN NONE?
0764 01 00      CMA 000
0764 27 00      BEB EX174
0764 01 CC02    CMA 000 EX174
0771 27 00      BEB EX174
0773 00 CB40    JBR INDEC GET FIRST FILE NUMBER
0774 24 00      BEC BANC
0770 0E 00EE    BRANE LBI 1000000 ILLEGAL NUMBER
0770 0F AD      EXONES CLR PUTSBN
0770 7E 00EE    JBP EPOBUN

0780 9F 92      MANK STI LORAND
0782 27 F4      BEB BANC NO FILE 0
0784 30 1F      BE1 FILE 1=ZERO OFFSET
0786 9F 92      STI LORAND
0788 BA CC11    LDA LSTTAN
0788 01 70      CMA 0-- RANGE DESIRED?
0788 26 10      BEE MORGAN
078F 00 CB40    JBR INDEC GET SECOND FILE NUMBER
0792 25 E4      BEC BANC
0794 9F 94      STI NIBAND SAVE UPPER LIMIT
0794 27 E0      BEB BANC
0798 30 1F      BE1 FILE 1=ZERO OFFSET
0798 9F 94      STI NIBAND
079C 96 92      RALOP LDA LORAND
079E 06 91      LDA LORAND+1
07A0 91 94      CMA NIBAND
07A2 22 0F      BHI NIBAND YES
07A4 23 04      BLO MORGAN NO
07A6 01 93      CMA NIBAND+1
07A8 22 0F      BHI NIBAND YES
07AA 96 92      RALOP LDA LORAND
07AC 0A 93      LDA LORAND+1
07AE 50        ASLJ
07AF 4F        RCLJ
07B0 50        RCLJ
07B1 4F        RCLJ
07B2 50        RCLJ
07B3 4F        RCLJ
07B4 50        ASLJ
07B5 4F        RCLJ
07B6 00 03      ADDO SECOR+1
07B8 9F 02      MCA SEC10
07BA 01 04      CMA BUNFED POST END OF DIR?
07BC 23 06      BLO MANK
07BE 22 00      BHI BANC
07C0 01 05      CMA BUNFED+1
07C2 24 04      BEB BANC
07C4 97 00      STAA TEMP1
07C6 07 0F      STAB TEMP1+1
07C8 9E 00      LBI TEMP1
07CA 00 04      STI 0,1
07CC 20 0A      BE1 BANC DELETED FILE
07CE 00 33      PSR RPTCPY
07D0 70 0142    LST LST77L0 LIST WITHOUT COPY?
07D2 2A 04      BEE SKP77L
07D4 00 00      BEB MVSFC3 MOVE 1010 FCBS
07D6 00 7E      PSR CPV77L
07D8 0C 93      SKP77L INC LORAND+1
07DA 26 0F      BEE RALOP
07DC 0C 92      INC LORAND
07DE 20 00      BEA RALOP

```

```

* MOVE FROM DIR TO FCBS
*
07E1 0E 1147    MVSFC3 LBI MVSFC3+1 MOVE TO HERE
07E4 00 03      BEB
07E6 0E 1207    MVSFC3 LBI MVSFC3+1 HERE ALSO
07E9 9F 0E      MVSFC3 STI TEMP4 TO POINTER
07EB 9F 00      LBI TEMP1 POINTS AT DIR
07ED 9F 0C      STI TEMP3 FROM POINTER
07EF C6 00      LDA 011
07F1 9F 0C      MVSFC3 LBI TEMP3
07F3 06 04      LDA 0,1
07F5 30 01      BE1
07F7 9F 0C      STI TEMP3
07F9 9F 0E      LBI TEMP4
07FB A7 04      STAA 0,1
07FD 30 01      BE1
07FF 9F 0E      STI TEMP4
0801 3A        BEC0
0802 26 ED      BEE MVSFC2
0804 3F        RTS

```

```

* REPORT COPY OF FILE
*
* ENTER -- TEMP1-COPY
*
0805 00 C074    RPTCPY JBR PCOLR ALL ON ESCAPE

```

```

0808 96 90      LDA SECORV
080A 97 9C      STAA 001VE SET CLARITY DRIVE
080C 9E 00      LBI TEMP1 POINT AT SOURCE
080E 0A 01      LDA 01
0810 97 03      STAA 0007L8 SHORT ENTRY HERE
0812 9F 90      STI TEMP3
0814 00 00F4    JBR OUTMAN TERMINAL
0817 00 AC      STI MVSFL8
0819 27 0F      BEB MVSFC3
081B 9E 90      LBI TEMP3
081D 96 03      LDA OUTTAN
081F 07 0002    STAB OUTTAN
0822 00 00F4    JBR OUTMAN
0824 06 FF      LDA 0007
0827 07 0002    STAB OUTTAN BACK TO TERMINAL
082A 70 0142    MVSFC3 STI LST77L8 REPORT ONLY?
082B 26 27      BEE MVSFC3
082F 0E 00C0    LBI 0100010 to drive message
0832 96 9E      LDA MVSFCV
0834 70 0141    STI 0107L8
0837 27 05      BEB 1200000
0839 0E 1027    LBI 0110000
083E 96 90      LDA SECORV
083F 34 02      TOBPM PSMA
0840 00 0075    JBR PMTA
0843 35 02      MCA 0
0845 00 30      ADDO 0*0 MAKE ABC11
0847 7E C010    JBP PUTCHN

```

```

* ADD 0 TO TEMP1
*
08D1A LBA0 016 COMPX NUMBER
08D0F1 ADDO TEMP1+1
08D10 STAB TEMP1+1
08D11 BEC ADD07
08D12 INC TEMP1
08D13 LBI TEMP1
08D14 RETRMS RTS

```

```

*****

```

```

* COPY FILE BANC TO FILE DESFC
*
* CHECK IF FILE EXISTS AND FIT
*

```

```

08E7 0E 1207    CPV77L LBI 4DESFC+4
08EA 00 00C3    JBR FIB77L FILE EXISTS ON UNIT?
08EB 27 50      BEB MVSIST
08ED 0E 0F17    LBI MVSISTH REPORT EXISTS
08EF 00 0075    JBR PMTA
08F1 70 0141    STI 0107L8 KILL SOURCE?
08F4 27 10      BEB MVSILL
08F7 70 0143    STI MVSFL8
08FA 26 00      BEE YESPRN
08FB 0E 0F49    LBI MVSILLN
08FD 00 0075    JBR PMTA REPORT
08FF 20 00      BEA MVSFMT

```

```

08G7 0E 0F43    YESPRN LBI 0MVSFC3 KILL PROMPT
08G9 00 04C0    JBR 4MVSFC3
08GB 26 10      BEE MVSILL
08GD 7E 042C    REPORT JBP KILLIT

```

```

08H2 70 013F    MVSILL STI 0107L8 ALWAYS DELETE EXISTING?
08H5 26 0F      BEE MVSILL
08H7 0E 0F23    LBI 0MVSFC3
08H9 00 04C0    JBR 4MVSFC3
08HB 27 07      BEB MVSILL
08HD 01 46      MVSFC3 CMA 0*F
08HE 26 C3      BEE MVSILL OTHER-4
08HF 7E 000F    JBP 0011 FILE
08I6 0E 1703    MVSILL LBI MVSFC3
08I9 06 0C      LDA 012
08IB 07 04      STAA 0,1
08IC 00 0404    JBR FIB8
08IE 06 00 24    BEE MVSFC3
08I9 06 00 24    LDA 024,2
08I9 07 04      STAA 0,1
08I9 07 00 13    LDA 21,2
08I9 03 00 16    LDA 22,1
08I9 00 07      ADDO REMAIN+1
08I9 07 97      STAB REMAIN+1
08I9 9F 96      MCA REMAIN
08I9 9F 96      STAA REMAIN
08I9 9F 00      BE1ST LBI TEMP1 POINT AT SRC DIR
08I9 06 97      LDA REMAIN+1 WILL FILE FIT?
08I9 00 0C      SUBB 12,1
08I9 07 97      STAB REMAIN+1
08I9 96 96      LDA REMAIN

```

```

0000 A2 00      S00A 11,1
0001 97 96      STAA 0,1
0003 24 00      BCC 0,1
0005 0F A8      CLR 0,1
0007 0E 0F33    LDI 0,1
000A 7E 0AEE    JMP 0,1

```

```

* READ SOURCE FILE
*
* CHECK RECORD NUMBERS AND EOF
*

```

```

0003 0E 1143    00LIT LDI 0,0
0004 04 01      LDAA 0,1
0005 A7 04      STAA 0,1
0006 00 040A    JBR 0,1
0007 27 03      BEQ 0,1
0009 7E 0A39    JMP 0,1
000C 0F 90      RECV2 CLR 0,1
000E 0F 99      CLR 0,1
0010 0F A9      CLR 0,1
0012 0F 00      CLR 0,1
0014 0F 04      CLR 0,1
0016 04 FF      LDAA 0,1
0018 A7 00 30   STAA 0,1
0019 0E 04      LDBR 0,1
001A 0F 0E      STI 0,1
001B 0E 1143    LDBR 0,1
001C 00 040A    JBR 0,1
001D 26 41      BNE 0,1
001F AE 00 42    LDBR 0,1
0020 0E 90      CPD 0,1
0021 27 20      BEQ 0,1
0022 0E 99      INC 0,1
0023 0E 90      INC 0,1
0024 0E 90      CPD 0,1
0025 27 16      BEQ 0,1
0026 00 0130    TST 0,1
0027 26 11      BNE 0,1
0028 78 0144    TST 0,1
0029 27 1E      BEQ 0,1
002A 00 90      TST 0,1
002B 26 33      BNE 0,1
002C 04 99      LDAB 0,1
002D C1 01      CPD 0,1
002E 26 20      BNE 0,1
002F 0E 90      STI 0,1
0030 0E 0E      RECV 0,1
0031 07 04      STAA 0,1
0032 20 01      INI 0,1
0033 0F 0E      STI 0,1
0034 0E 0A      CPD 0,1
0035 26 05      BNE 0,1
0036 00 00      TST 0,1
0037 27 32      BEQ 0,1
0038 20 40      BNA 0,1

```

```

0039 0F A8      DECODE CLR 0,1
003A 0E 10F9    LDI 0,1
003B 7E 0AEE    JMP 0,1

```

```

0039 A6 01      CHECKF LDAA 0,1
003A 01 00      CPD 0,1
003B 27 00      BEQ 0,1
003C 01 00      CPD 0,1
003D 26 17      BNE 0,1
003E 78 0130    TST 0,1
003F 27 12      BEQ 0,1
0040 20 4E      BNA 0,1
0041 04 04      CLDRC LDAA 0,1
0042 07 04      STAA 0,1
0043 07 04      STAA 0,1
0044 78 0144    TST 0,1
0045 26 00      BNE 0,1
0046 00 040A    JBR 0,1
0047 27 03      BEQ 0,1
0048 7E 0A39    JMP 0,1

```

```

* WRITE DESTINATION FILE
*

```

```

004C 00 A0      H0002 TST 0,1
004D 24 30      BNE 0,1
004E 0E 1203    CPD 0,1
004F 04 02      LDAA 0,1
0050 07 A0      STAA 0,1
0051 07 04      STAA 0,1
0052 00 040A    JBR 0,1
0053 26 03      BNE 0,1
0054 0F 04      CLR 0,1

```

```

0070 04 FF      LDAA 0,1
0071 A7 00 30   STAA 0,1
0072 04 1152    LDAA 0,1
0073 A7 0F      STAA 0,1
0074 04 1150    LDAA 0,1
0075 27 05      BEQ 0,1
0076 04 02      LDAA 0,1
0077 A7 00 17   STAA 0,1
0078 78 0140    TST 0,1
0079 26 12      BNE 0,1
007A 04 115C    LDAA 0,1
007B A7 00 19   STAA 0,1
007C 04 1150    LDAA 0,1
007D A7 00 1A   STAA 0,1
007E 04 115E    LDAA 0,1
007F A7 00 1B   STAA 0,1
0080 04 04      LDAA 0,1
0081 0E 1203    LDI 0,1
0082 00 040A    JBR 0,1
0083 26 00      BNE 0,1
0084 E4 00 1E    LDAB 0,1
0085 C1 01      CPD 0,1
0086 22 1A      BNE 0,1
0087 25 04      BLO 0,1
0088 00 AF      TST 0,1
0089 27 14      BBO 0,1
008A 0F 0E      BNA 0,1
008B 04 9A      LDAB 0,1
008C 04 90      LDAA 0,1
008D A7 00 1E    STAB 0,1
008E 07 0F      INC 0,1
008F 0E AE      INC 0,1
0090 0E 1122    LDI 0,1
0091 7E 0AEE    JMP 0,1
0092 07 9A      LDBR 0,1
0093 E4 00 1F    LDAB 0,1
0094 07 90      STAB 0,1
0095 30 01      JBR 0,1
0096 0E AE      CPD 0,1
0097 26 C4      BNE 0,1
0098 00 A9      TST 0,1
0099 26 03      BNE 0,1
009A 7E 0AEE    JMP 0,1

```

```

* CHECK DELETE AND SECOND COPY
*

```

```

009B 0E 1203    H000E LDI 0,1
009C 04 04      LDAA 0,1
009D A7 04      STAA 0,1
009E 00 040A    JBR 0,1
009F 26 40      BNE 0,1
00A0 04 01      LDAA 0,1
00A1 07 04      STAA 0,1
00A2 0E 0F3F    LDI 0,1
00A3 00 0075    JBR 0,1
00A4 04 AC      TST 0,1
00A5 27 10      BEQ 0,1
00A6 04 A3      LDAA 0,1
00A7 07 CC22    STAA 0,1
00A8 0E 0F3F    LDI 0,1
00A9 00 0075    JBR 0,1
00AA 04 FF      LDAA 0,1
00AB 07 CC22    STAA 0,1
00AC 78 0147    TST 0,1
00AD 27 10      BEQ 0,1
00AE 00 4350    LDI 0,1
00AF 04 59      LDAA 0,1
00B0 0C 120F    CPD 0,1
00B1 24 03      BNE 0,1
00B2 01 1291    CPD 0,1
00B3 27 09      BEQ 0,1
00B4 0F 120F    YESDEC STI 0,1
00B5 07 1291    STAA 0,1
00B6 7E 0AEE    JMP 0,1

```

```

00B7 78 0144    H000E TST 0,1
00B8 27 0C      BEQ 0,1
00B9 0E 1143    KILLIT LDI 0,1
00BA 04 0C      LDAA 0,1
00BB A7 04      STAA 0,1
00BC 00 040A    JBR 0,1
00BD 26 01      BNE 0,1
00BE 39 00      H001P RTD 0,1

```

```

* ERROR REPORT
*
* SEPARATE DECODE FOR FILE OPERATIONS
* AND SECTOR READ WRITE, PRINTS NOT
* RESSAID WITHOUT ERROR, ETC.

```

```

0039 9F 0E ERROR STI TEMP4 SAME FOR OPTERR
0030 A6 01 L00A 1,1 SET FMS ERROR
0030 0E 10F9 L01 0PCECTR
0040 01 19 C0FA 025
0042 27 23 B0E0 E00002
0044 0E 1073 L00 000YEAR
0047 01 10 C0FA 016
0049 27 1E B0E0 E00002
0040 0E 1003 L01 0PPTERR
004E 01 00 C0FA 011
0050 27 17 B0E0 E00002
0052 0E 106A L01 0PPTERR
0053 01 04 C0FA 04
0057 27 10 B0E0 E00002
0059 0C AE INC P0T0AC PRINT TRK/SEC
0050 0E 10A7 L01 0PPTERR
005E 01 0A C0FA 010
0060 27 07 B0E0 E00002
0062 0E 10CE L01 0PPTERR
0063 01 09 C0FA 09
0067 26 03 B0E0 E00002
0069 7E 0AEE E00002 JNP
006C 9E 0E RPTERR L01 TEMP4
006E 00 C03F J0R RPTERR
0071 00 C024 J0R PC0LF
0074 00 A8 E0112 TEST P0T0AC
0076 27 37 B0E0 E011 SKIP DRIVE PRIOT?
0070 0E 00C3 L01 000000
0070 00 0075 J0R P0MTA
007E 9E 0E L01 TEMP4
0080 A6 03 L00A 3,1 GET ERROR DRIVE
0082 0A 30 000A 0'0 NICE ASCII
0084 00 C010 J0R P0T0AC
0087 00 AE TEST P0T0AC READ/WRITE/FIND
0089 27 24 B0E0 E011
0080 0E 1039 L01 0TRACON
008E 99 0075 J0R P0MTA
0091 9E 0E L01 TEMP4
0093 C6 1E L00A 000E
0095 00 C03A J0R ADD01
0090 00 C03C J0R 00T0E1 PRINT TRACE
0090 00 0070 J0R 00T0E0
009E 30 01 001
00A0 00 C03C J0R 00T0E1 PRINT SECTOR
00A3 00 A0 007 010'0 READING SEC BIT?
00A5 27 00 B0E0 E011
00A7 0E 00F3 L01 000000 SKIP REMAINING BIT?
00AA 00 04C0 J0R 000000
00AB 27 10 B0E0 RET007
00AB 00 0403 J0R 000000 CLOSE OFF BITS
00B2 9A A4 L00A P0MT0N
00B4 87 C009 ST0A PAUSE
00B7 00 A4 007 00T0L
00B9 26 04 B0E0 000000 SOME FILES COPIED?
00B7 0E 00F7 L01 000000
00B0 00 C01E J0R P0T0AC
00C1 9A A3 B0000 L00A 00T00
00C3 07 C022 ST0A 00T00 RESTORE OLD OUT SWITCH
00CA 30 C024 J0R P00L5
00C7 7E C003 JNP 00000
00CC 39 RET007 010
00C9 9F 0E B0000 STI TEMP4 SAME FC0
00C7 A6 01 L00A 1,1 SET R/W ERROR
00D1 0E 1073 L01 000YEAR
00D4 40 0010 L01 00000
00D5 25 17 B0C0 E00000 BIT 7
00D7 0E 1003 L01 0PPTERR
00D8 40 0010 L01 00000
00D0 25 11 B0C0 E00000 BIT 4
00D0 0C AE INC P0T0AC PRINT TRK/SEC
00D0 0E 10A7 L01 0PPTERR
00E2 40 0010 L01 00000
00E3 25 09 B0C0 E00000 BIT 5
00E5 0E 1000 L01 0PPTERR
00E6 40 0010 L01 00000
00E7 25 03 B0C0 E00000 BIT 4
00E7 0E 10CE L01 0PPTERR
00EE 00 C01E J0R P0T0AC
00F1 7E 0074 JNP E0112

```

```

* FORMATTED OUTPUT OF FILE
* NAME, SIZE, AND DATE
*
* ENTER --- 1-1111 TO OUTPUT
*
*

```

```

00F6 0A 30 000A 0'0 MAKE ASCII
00F8 00 70 000A 0'0 PERIOD
00FA 0A 2E 000A 0'0 PERIOD
00FC 00 74 000A 0'0 PERIOD
00FE C6 00 000A 0'0 PERIOD
0000 00 75 000A 0'0 PERIOD
0002 0A 2E 000A 0'0 PERIOD
0004 00 4C 000A 0'0 PERIOD
0006 C6 03 000A 0'0 PERIOD
0008 00 60 000A 0'0 PERIOD
000A 00 A3 000A 0'0 PERIOD
000C 2A 05 000A 0'0 PERIOD
000E C6 06 000A 0'0 PERIOD
0010 00 C03A J0R 00000
0012 9F 0C 0010 0'0 PERIOD
0014 C6 01 0010 0'0 PERIOD
0016 00 C039 J0R 00000
0018 9E 0C 0010 0'0 PERIOD
001C 30 01 0010 0'0 PERIOD
001E 00 A3 0010 0'0 PERIOD
0020 2A 04 0010 0'0 PERIOD
0022 30 01 0010 0'0 PERIOD
0024 30 01 0010 0'0 PERIOD
0026 9F 0C 0010 0'0 PERIOD
0028 0A 20 0010 0'0 PERIOD
002A 00 4A 0010 0'0 PERIOD
002C A6 02 0010 0'0 PERIOD
002E 01 09 0010 0'0 PERIOD
0030 22 00 0010 0'0 PERIOD
0032 3A 02 0010 0'0 PERIOD
0034 0A 20 0010 0'0 PERIOD
0036 00 3A 0010 0'0 PERIOD
0038 35 02 0010 0'0 PERIOD
003A 3F 00 0010 0'0 PERIOD
003C 00 29 0010 0'0 PERIOD
003E 00 30 0010 0'0 PERIOD
0040 9E 0C 0010 0'0 PERIOD
0042 A6 01 0010 0'0 PERIOD
0044 27 04 0010 0'0 PERIOD
0046 01 00 0010 0'0 PERIOD
0048 25 02 0010 0'0 PERIOD
004A 0A 00 0010 0'0 PERIOD
004C 0E 10AC 0010 0'0 PERIOD
004E 4A 00 0010 0'0 PERIOD
0050 27 00 0010 0'0 PERIOD
0052 30 01 0010 0'0 PERIOD
0054 30 01 0010 0'0 PERIOD
0056 30 01 0010 0'0 PERIOD
0058 20 03 0010 0'0 PERIOD
005A C6 03 0010 0'0 PERIOD
005C 00 10 0010 0'0 PERIOD
005E 30 01 0010 0'0 PERIOD
0060 00 0F 0010 0'0 PERIOD
0062 9E 0C 0010 0'0 PERIOD
0064 A6 03 0010 0'0 PERIOD
0066 3F 00 0010 0'0 PERIOD
0068 97 0F 0010 0'0 PERIOD
006A 0F 0E 0010 0'0 PERIOD
006C 00 000E 0010 0'0 PERIOD
006E 7E C039 J0R 00000
0070 0A 20 0010 0'0 PERIOD
0072 7E C010 J0R 00000
0074 C6 0F 0010 0'0 PERIOD
0076 A6 04 0010 0'0 PERIOD
0078 2A 02 0010 0'0 PERIOD
007A 0A 20 0010 0'0 PERIOD
007C 01 04 0010 0'0 PERIOD
007E 27 07 0010 0'0 PERIOD
0080 00 0F 0010 0'0 PERIOD
0082 30 01 0010 0'0 PERIOD
0084 5A 00 0010 0'0 PERIOD
0086 2A 0F 0010 0'0 PERIOD
0088 39 00 0010 0'0 PERIOD
008A C6 0F 0010 0'0 PERIOD
008C A6 04 0010 0'0 PERIOD
008E 2A 02 0010 0'0 PERIOD
0090 0A 20 0010 0'0 PERIOD
0092 01 04 0010 0'0 PERIOD
0094 27 07 0010 0'0 PERIOD
0096 00 0F 0010 0'0 PERIOD
0098 30 01 0010 0'0 PERIOD
009A 5A 00 0010 0'0 PERIOD
009C 2A 0F 0010 0'0 PERIOD
009E 39 00 0010 0'0 PERIOD
00A0 0E C040 STATUS L01 00C0
00A2 9A 9E L00A 00000
00A4 07 03 ST0A 3,1
00A6 0A 03 L00A 03
00A8 A7 00 1F ST0A 01F,1
00AA 0F 00 1E CL0 01E,1
00AC 0A 09 L00A 09 READ COMMAND

```

```

* CHECK STATUS OF REGISTRATION BLOCK
* FINDS SECTIONS REMAINING FROM 010
* CHECKS WRITE PROTECT BY 001100
* TO TRACK ZERO SECTOR FOUR.
*
* ENTER --- 0000-0000 TO TEST
* ENTER --- 0011-0011 IF OK

```



```

009A A7 04      STAA 0,1
009C 1A 10      SETI
009E 00 0000    JSR    FRS      STOP INTERRUPTS
00A1 27 03      BEQ    B10K   READ S10
00A3 7E 0A00    EROU0S JMP    FRS00   UNABLE TO READ S10?

```

```

00A6 06 C97E    STROK L0A0 FCR=310
00A9 01 FF      C0P4 00FF EXTENDED DIRECTORY FLAG
00AB 26 02      BNE    NOE11
00AD 0C AF      INC    E17FL0 SET EXTENDED MODE
00AF 06 04      NOE11 L0A0 04
00B1 07 00 1F   STMA 01F,1 COPY S10 TO SECTION 4
00B4 06 06      L0A0 010
00B6 A7 04      STAA 0,1
00B8 1A 10      SETI
00BA 00 0000    JSR    FRS      STOP INTERRUPTS
00BC 2A E4      BNE    EROU0S
00BE 0E 00 A1   L01 0A1,1 SECTIONS REMAINING
00C2 0F 06      STI    REMAIN
00C4 39          RTS

```

```

* FIND FILE IN DIRECTORY
*
* ENTER --- I-ENTRY TO FIND
* EXIT --- EQUAL FLAG IF NOT FOUND
*          TEMP2=MATCH IF FOUND
*

```

```

00C5 0F 00      F00F11 STI    TEMP2
00C7 0E 00      L01 0E0010
00C9 0C 02      C01 00C010 E0FF0 B10?
00CB 27 00      BEQ    BE1006
00CD 0F 00      F00F11 STI    TEMP2
00CF 0F 0E      F00F0P STI    TEMP4
00D1 0F 00      L01 0E0010
00D3 0F 0C      STI    TEMP3
00D5 0C 00      L0A0 011
00D7 0E 0C      F00F12 L01 0E0010
00D9 A6 04      L0A0 0,1
00DB 30 01      L01 0E0010
00DD 0F 0C      STI    TEMP3
00DF 0E 0E      L01 0E0010
00E1 A1 04      C0P4 0,1
00E3 26 09      BNE    NOTFND
00E5 30 01      L01 0E0010
00E7 0F 0E      STI    TEMP4
00E9 5A 00      BEQ    BE1006
00EB 26 E0      BNE    F00F12
00ED 3C          INC0
00EF 39          RTS      FLAG FOUND
                                MATCH IN TEMP2

```

```

00EE 06 00      NOTFND L0A0 TEMP2+1
00F0 00 10      ADD0 010
00F2 07 00      STAA TEMP2+1
00F4 24 02      BCC    NOTFND
00F6 0C 0A      INC    TEMP2
00F8 0E 0A      NOTFND L01 0E0010
00FA 00 A7      STI    S00LFL
00FC 27 04      BEQ    NOE11
00FE 0C 04      C01 00FF0
0100 20 02      B0A 10100L

```

```

00C2 0C 02      NOTSIN C01 00C010
00C4 26 C9      ISIN0L BNE    F00F0P
00C6 39          RETR00 RTS      DONE?
                                TEMP FLAG NOT FOUND

```

```

* RECOVER FILE FROM TRACK-SECTION BY
* SIMULATING SOURCE FILE BEING OPEN
* AND OPENING DESTINATION FILE.
*

```

```

00C7 0E 1303    RECOVER L01 0E0010
00C9 0F 04      STI    00FF0
00CB 7C 0140    INC    00F0FL0 UPDATE TO CURRENT
00CD 00 33      B0R    GETBYT
00C1 04 07      ADD0 07
00C3 07 1146    STAA 00F0C0+3 M10E
00C5 00 2C      B0R    GETBYT
00C7 07 1143    STAA 00F0C0+440 LINK TRACK
00C9 00 27      B0R    GETBYT
00CB 07 1104    STAA 00F0C0+441 LINK SECTION
00CD 0E 1203    L01 00F0C0
00C1 00 C020    JSR    00F0JL DESTINATION FILE
00C3 25 20      B0C    SYNT14
00C5 06 06      L0A0 06
00C7 06 C033    JSR    SETBYT
00C9 0E 1143    L01 00F0C0
00CB 0F 01      CLP 1,1
00C5 06 01      L0A0 01
00C3 07 02      STAA 2,1 SET READ STATUS
00C1 0F 0F      CL0 15,1
00C3 0F 0F 17  CL0 25,1

```

```

00C3D 0F 00 22  CL0 622,1 NEXT BYTE TO READ
00C3E 00 000C   JSR    RECDV2 COPY FILE
00C40 7E 0A0F    JMP    EXIT

```

```

00C44 00 C042    GETBYT JSR    GETBYT
00C47 25 07      B0C    SYNT14
00C49 0F 00      STI    TEMP5 16 BITS
00C4B 06 01      L0A0 TEMP5+1 L01
00C4D 50          ST0
00C4E 76 06      BNE    RETR00
00C50 7E 0000    SYNT14 JMP    SYNTA3 STRICT FORMAT

```

* MESSAGES

```

00C53 20 63 4F 50  HLPH03 FCC  ' COPY A00L 0,1 .END A0'
00C57 39 20 41 44
00C5B 1E 4C 20 30
00C5F 2C 31 20 2E
00C63 63 10 44 20
00C67 41 42
00C69 20 20 4E 4F  FCC  ' (normal)'
00C6B 72 60 61 6C
00C71 29
00C73 04
00C75 20 43 4F 50  HLPH02 FCC  ' COPY FE 1,2 12 2-7 21'
00C77 59 20 46 15
00C79 20 31 2C 32
00C7B 20 31 32 20
00C7D 32 20 37 20
00C7F 32 31
00C81 20 20 66 69  FCC  ' (file 0)'
00C83 6C 63 20 23
00C85 29
00C87 04
00C89 20 43 4F 50  HLPH03 FCC  ' COPY R 0 20 0 1 FILE '
00C8B 59 20 32 20
00C8D 30 20 32 61
00C8F 20 30 20 31
00C91 2E 66 49 4C
00C93 45 20
00C95 20 20 73 63  FCC  ' (sector)'
00C97 63 74 4F 72
00C99 29
00CA1 04
00CA3 61 20 20 41  OPT003 FCC  'A Alphabetize files'
00CA5 6C 70 40 61
00CA7 62 63 74 69
00CAB 74 63 20 66
00CAC 69 6C 63 73
00CC7 04
00CC9 43 20 20 61
00CCA 6C 6C 4F 77
00CCB 20 13 4F 72
00CCD 72 75 70 74
00CCF 20 66 69 6C
00CD1 63 73
00CD3 20 20 64 61
00CD5 6E 67 63 72
00CD7 29
00CE7 04
00CE9 44 20 20 63
00CEB 6F 70 79 20
00CED 6F 6E 6C 79
00CEF 20 66 69 6C
00CF1 63 73 20
00CF3 77 69 74 60  FCC  'with source data'
00CF5 20 6E 63 77
00CF7 63 72 20 44
00CF9 61 74 63
00D01 04
00D03 45 20 20 64
00D05 63 6C 63 74
00D07 63 20 43 70
00D09 69 73 74 69
00D0B 6E 67 20 66
00D0D 69 6C 63 20
00D0F 66 72 6F 60  FCC  'free destination'
00D11 20 64 63 73
00D13 74 69 6E 61
00D15 74 69 6F 6E
00D17 04
00D19 46 20 20 63
00D1B 6F 70 79 20
00D1D 62 79
00D1F 20 44 69 6C  FCC  'File number'
00D21 63 20 6E 73
00D23 60 62 63 73
00D25 04
00D27 48 20 20 44  FCC  '
00D29 6F 6C 6C 20  FCC  'E Kill source '
00D2B 6F 6C 6C 20
00D2D 73 6F 73 72
00D2F 63 63 20
00D31 64 75 70 6C  FCC  'duplicate file'

```

```

0026 49 63 61 74
0043 45 20 66 69
0060 46 63
0076 20 20 64 61
0093 46 67 63 72
0110 29
0127 04
0144 46 20 20 46
0161 66 73 74 20
0178 77 69 74 68
0195 46 73 70 20
0212 63 67 70 79
0229 04
0246 46 20 20 63
0263 46 70 70 20
0280 66 69 66 63
0297 73 70 66 57
0314 74 20
0331 46 20 66
0348 65 73 74 69
0365 46 61 74 69
0382 66 66
0399 74
0416 46 20 20 74
0433 73 72 66 20
0450 46 64 64 20
0467 64 63 66 61
0484 73 66 74 73
0501 04
0518 50 20 20 30
0535 72 66 60 70
0552 71 20 62 63
0569 66 66 72 63
0586 20 63 67 70
0603 79 69 66 67
0620 20
0637 66 69 66 63
0654 04
0671 52 20 20 32
0688 63 63 67 76
0705 65 72 20 66
0722 72 66 60 20
0739 76 72 68 20
0756 73 63 63
0773 04
0790 53 20 20 33
0807 63 63 67 66
0824 64 20 63 67
0841 70 79
0858 04
0875 33 20 20 33
0892 72 63 20 63
0909 73 72 63
0926 46 71 20 66
0943 62 63 70 20
0960 64 61 74 63
0977 04
0994 57 20 20 57
1011 61 69 74 20
1028 66 66 72 20
1045 64 69 73 68
1062 20 63 68 61
1079 66 67 63
1096 04
1113 34 20 20 34
1130 61 70 20 73
1147 66 73 72 63
1164 63 20 66 69
1181 63 63
1198 20 20 64 61
1215 66 67 63 72
1232 29
1249 04
1266 44 61 66 67
1283 63 72 66 70
1300 73 20 66 70
1317 74 69 66 66
1334 20 73 63 66
1351 65 62 71 63
1368 64 20
1385 07 04
1402 41 72 63 20
1419 79 66 73 20
1436 73 73 72 63
1453 20 20 39 28
1470 46 20 34 20
1487 04
1504 20 46 46 20
1521 20 20
1538 04
1555 20 39 43 33
1572 20 20
1589 04
1606 42 41 44 20

```

```

FCC ' ideagel'
FCB 4
FCC 'L List without copy'
FCB 4
FCC 'M copy files not'
FCC 'on destination'
FCB 4
FCC 'D turn off defaults'
FCB 4
FCC 'P Prompt before copying'
FCC 'file'
FCC 'B Recover from trs-sec'
FCB 4
FCC 'B Second copy'
FCB 4
FCC 'U Use current files date'
FCB 4
FCC 'M Wait for disk change'
FCB 4
FCC 'I Load source file'
FCC ' (danger)'
LSTOPT FCB 4
DANGER FCB 'Dangerous option selected'
SURE FCB 7,4
FCC 'Are you sure (Y/N):'
FCB 4
FCC 'NO'
FCB 0
YES FCB 'YES'
FCB 4
DANGER FCB 'BAD ENTRY:'

```

```

0692 45 46 59 32
0709 59 58 20 20
0726 04
0743 41 41 54 45
0760 20 42 41 44
0777 2A 20 20 20
0794 04
0811 43 46 46 46
0828 57 20 42 41
0845 11 20 44 41
0862 54 45 53 20
0879 20 39 28 46
0896 29 34 20
0913 04
0930 20 74 66
0947 20 41 72 69
0964 76 63 20 23
0981 04
0998 53 49 46 47
1015 46 45 20
1032 43 46 50 59
1049 20 42 59 20
1066 46 55 48 42
1083 43 32 20 46
1100 46 54
1117 20 41 46 46
1134 46 57 45 44
1151 04
1168 49 46 46 45
1185 47 41 46 20
1202 44 49 52 45
1219 43 54 46 32
1236 39 20 46 33
1253 48 42 43 32
1270 04
1287 43 68 61 66
1304 67 63 20 64
1321 49 73 68 20
1338 70 72 63 73
1355 72 20 68 63
1372 79 34 20
1389 04
1406 20 20 46 69
1423 46 63 20 63
1440 78 69 73 74
1457 73
1474 04
1491 20 20 41 63
1508 41 66 63 74 63
1525 20 66 72 69
1542 67 69 66 61
1559 46 20 20 59
1576 28 46 28 46
1593 29 38 20
1610 04
1627 20 26 20 40
1644 69 66 66 63
1661 64
1678 04
1695 46 66 66 63
1712 46 66 66 63
1729 46 66 66 63
1746 46 66 66 63
1763 46 66 66 63
1780 46 66 66 63
1797 46 66 66 63
1814 46 66 66 63
1831 46 66 66 63
1848 46 66 66 63
1865 46 66 66 63
1882 46 66 66 63
1899 46 66 66 63
1916 46 66 66 63
1933 46 66 66 63
1950 46 66 66 63
1967 46 66 66 63
1984 46 66 66 63
2001 46 66 66 63
2018 46 66 66 63
2035 46 66 66 63
2052 46 66 66 63
2069 46 66 66 63
2086 46 66 66 63
2103 46 66 66 63
2120 46 66 66 63
2137 46 66 66 63
2154 46 66 66 63
2171 46 66 66 63
2188 46 66 66 63
2205 46 66 66 63
2222 46 66 66 63
2239 46 66 66 63
2256 46 66 66 63
2273 46 66 66 63
2290 46 66 66 63
2307 46 66 66 63
2324 46 66 66 63
2341 46 66 66 63
2358 46 66 66 63
2375 46 66 66 63
2392 46 66 66 63
2409 46 66 66 63
2426 46 66 66 63
2443 46 66 66 63
2460 46 66 66 63
2477 46 66 66 63
2494 46 66 66 63
2511 46 66 66 63
2528 46 66 66 63
2545 46 66 66 63
2562 46 66 66 63
2579 46 66 66 63
2596 46 66 66 63
2613 46 66 66 63
2630 46 66 66 63
2647 46 66 66 63
2664 46 66 66 63
2681 46 66 66 63
2698 46 66 66 63
2715 46 66 66 63
2732 46 66 66 63
2749 46 66 66 63
2766 46 66 66 63
2783 46 66 66 63
2800 46 66 66 63
2817 46 66 66 63
2834 46 66 66 63
2851 46 66 66 63
2868 46 66 66 63
2885 46 66 66 63
2902 46 66 66 63
2919 46 66 66 63
2936 46 66 66 63
2953 46 66 66 63
2970 46 66 66 63
2987 46 66 66 63
3004 46 66 66 63
3021 46 66 66 63
3038 46 66 66 63
3055 46 66 66 63
3072 46 66 66 63
3089 46 66 66 63
3106 46 66 66 63
3123 46 66 66 63
3140 46 66 66 63
3157 46 66 66 63
3174 46 66 66 63
3191 46 66 66 63
3208 46 66 66 63
3225 46 66 66 63
3242 46 66 66 63
3259 46 66 66 63
3276 46 66 66 63
3293 46 66 66 63
3310 46 66 66 63
3327 46 66 66 63
3344 46 66 66 63
3361 46 66 66 63
3378 46 66 66 63
3395 46 66 66 63
3412 46 66 66 63
3429 46 66 66 63
3446 46 66 66 63
3463 46 66 66 63
3480 46 66 66 63
3497 46 66 66 63
3514 46 66 66 63
3531 46 66 66 63
3548 46 66 66 63
3565 46 66 66 63
3582 46 66 66 63
3599 46 66 66 63
3616 46 66 66 63
3633 46 66 66 63
3650 46 66 66 63
3667 46 66 66 63
3684 46 66 66 63
3701 46 66 66 63
3718 46 66 66 63
3735 46 66 66 63
3752 46 66 66 63
3769 46 66 66 63
3786 46 66 66 63
3803 46 66 66 63
3820 46 66 66 63
3837 46 66 66 63
3854 46 66 66 63
3871 46 66 66 63
3888 46 66 66 63
3905 46 66 66 63
3922 46 66 66 63
3939 46 66 66 63
3956 46 66 66 63
3973 46 66 66 63
3990 46 66 66 63
4007 46 66 66 63
4024 46 66 66 63
4041 46 66 66 63
4058 46 66 66 63
4075 46 66 66 63
4092 46 66 66 63
4109 46 66 66 63
4126 46 66 66 63
4143 46 66 66 63
4160 46 66 66 63
4177 46 66 66 63
4194 46 66 66 63
4211 46 66 66 63
4228 46 66 66 63
4245 46 66 66 63
4262 46 66 66 63
4279 46 66 66 63
4296 46 66 66 63
4313 46 66 66 63
4330 46 66 66 63
4347 46 66 66 63
4364 46 66 66 63
4381 46 66 66 63
4398 46 66 66 63
4415 46 66 66 63
4432 46 66 66 63
4449 46 66 66 63
4466 46 66 66 63
4483 46 66 66 63
4500 46 66 66 63
4517 46 66 66 63
4534 46 66 66 63
4551 46 66 66 63
4568 46 66 66 63
4585 46 66 66 63
4602 46 66 66 63
4619 46 66 66 63
4636 46 66 66 63
4653 46 66 66 63
4670 46 66 66 63
4687 46 66 66 63
4704 46 66 66 63
4721 46 66 66 63
4738 46 66 66 63
4755 46 66 66 63
4772 46 66 66 63
4789 46 66 66 63
4806 46 66 66 63
4823 46 66 66 63
4840 46 66 66 63
4857 46 66 66 63
4874 46 66 66 63
4891 46 66 66 63
4908 46 66 66 63
4925 46 66 66 63
4942 46 66 66 63
4959 46 66 66 63
4976 46 66 66 63
4993 46 66 66 63
5010 46 66 66 63
5027 46 66 66 63
5044 46 66 66 63
5061 46 66 66 63
5078 46 66 66 63
5095 46 66 66 63
5112 46 66 66 63
5129 46 66 66 63
5146 46 66 66 63
5163 46 66 66 63
5180 46 66 66 63
5197 46 66 66 63
5214 46 66 66 63
5231 46 66 66 63
5248 46 66 66 63
5265 46 66 66 63
5282 46 66 66 63
5299 46 66 66 63
5316 46 66 66 63
5333 46 66 66 63
5350 46 66 66 63
5367 46 66 66 63
5384 46 66 66 63
5401 46 66 66 63
5418 46 66 66 63
5435 46 66 66 63
5452 46 66 66 63
5469 46 66 66 63
5486 46 66 66 63
5503 46 66 66 63
5520 46 66 66 63
5537 46 66 66 63
5554 46 66 66 63
5571 46 66 66 63
5588 46 66 66 63
5605 46 66 66 63
5622 46 66 66 63
5639 46 66 66 63
5656 46 66 66 63
5673 46 66 66 63
5690 46 66 66 63
5707 46 66 66 63
5724 46 66 66 63
5741 46 66 66 63
5758 46 66 66 63
5775 46 66 66 63
5792 46 66 66 63
5809 46 66 66 63
5826 46 66 66 63
5843 46 66 66 63
5860 46 66 66 63
5877 46 66 66 63
5894 46 66 66 63
5911 46 66 66 63
5928 46 66 66 63
5945 46 66 66 63
5962 46 66 66 63
5979 46 66 66 63
6000 46 66 66 63

```


REVIEW

META LAB DAC1220

The Meta Lab DAC1220 is a two channel Digital to Analog Converter with 12 bit resolution. It is designed for use in computers having the SS50C bus and requires one thirty pin slot on the I/O bus. The DAC1220 is available assembled and tested for \$395 from Meta Lab, 6825 County Line Road, Longmont, Colorado 80501.

Our initial impression of the DAC1220 was one of quality. The 14.5cm by 12.5cm circuit board is solder masked and silk screened. The Molex connectors are gold, all components are top quality, and all ICs are in sockets. Multi-turn pots are used instead of the more common (and cheaper) single turn variety, for example. Considering the price of the DAC1220 we expected, and got, a professional quality board.

What is meant by 12 bit resolution in a digital to analog converter? This indicates the number of unique values between the minimum and maximum outputs of the DAC. No Virginia, not 12 steps, 4096 steps. If we were discussing a 4 bit DAC, there would be 16 steps. An 8 bit DAC would have 256 steps, etc. Thus the DAC1220 gives us 4096 steps from 0 volts to +5 volts, or 4096 steps from -5 volts to +5 volts. The choice of unipolar (0 to +5 volt) output or bipolar (-5 to +5 volt) output is up to the user and is easily changed with a DIP switch. As the two channels operate independently, either may be bipolar while the other is unipolar; or both may be bipolar; or both may be unipolar.

The DAC1220 gives the user several options regarding output update. Each channel may be set to operate completely independently of the other, in which case the output voltage will change when a new digital value is given to it. Also either channel may be slaved to the other. Suppose we make channel A the slave of channel B, in this case feeding a new digital value to channel A will result in no change at its output until a new digital value is fed to channel B, then both channels will update together. When one channel is slaved to the other, their output voltage values are still independent of each other. The DAC1220 also provides a third output update option which, if used, requires a read after write to give a means of "handshaking". If this mode is used, feeding data to the DAC channel will cause no change in the output voltage value. When the program attempts to read the DAC1220, the output updates. This can be very useful when you are using the DAC1220 to feed a device that can only accept data at a limited rate. The device can generate an interrupt when ready for new data and the interrupt routine has only to read the DAC1220 to effect the update.

The Meta Lab DAC1220 has a very accurate internal reference, 5.0 volts \pm .5%. This reference voltage is required by the DAC chips and insuring its accuracy means increased accuracy of the output voltages. DAC chips are most accurate at their extremes and tend to lose accuracy towards mid-range. The precision reference supply helps to hold this error to a minimum. This internal reference is many times more accurate than the typical three terminal regulator. However, if it still isn't good enough for your application, either or both channels of the DAC1220 may be provided with an external reference. The external reference input is also used when the DAC1220 is to act as a four quadrant multiplying DAC in such applications as computer controlled gain. In this mode the input

signal is applied as the external reference of the appropriate channel.

All digital to analog converters output glitches (or spikes) whenever the output is changed and Meta Lab's DAC1220 is no exception. However they have provided a means of compensating for this characteristic whenever the DAC1220 is used in an application sensitive to these glitches. Using a DAC to control a CRT graphic display could have unpredictable results during glitch time. The DAC1220 has an UNBLANK output to notify the driven device when the output is unstable; which could be used to keep a CRT display from smearing due to glitches.

To the programmer the DAC1220 is four contiguous addresses in the I/O space of the computer. When the four addresses are read, it looks like an unequipped port; when they are written the output voltages of the DAC1220 change. The DAC1220 requires no initialization itself, but the device it is used to drive will in all probability need to be initialized. If you are driving an XY plotter, the pen must be located at a known point. The same is true of the cursor in a CRT graphic display and most other devices. We used the DAC1220 to drive a MOOG music synthesizer and were often greeted with a very LOUD random tone when the system was powered up. One minor complaint regarding programming the DAC1220 is that data is left justified, which can be confusing initially. Since we are dealing with byte wide machines, loading a value into the DAC requires that either an accumulator be stored at DAC and DAC+1 or that a sixteen bit register be stored at DAC. The DAC1220 is twelve bits wide on each input on the digital side, with the data left justified the least significant four bits are discarded. Many human brains attempt to right justify the data and discard the most significant four bits. Luckily it is not too difficult to mentally make then adjustment to left justified data. The DAC1220 is double buffered so the output voltage does not change until both bytes of the channel have been updated; which makes life easier regardless of what device you connect to the DAC. As the DAC does not require initialization, it can be painlessly accessed from any language with PEEK and POKE type capability. There are no status registers to be read and tested, no assembly language subroutines to be called. Just put the desired value into the address where the DAC channel is located. This is one of the easiest devices ever for the programmer to control, regardless of the language used. However it would be even easier if data were right justified.

Meta Lab warns in the documentation on the DAC1220 that it will not work on older SS50 bus computers unless the \pm 12 volt supplies have been upgraded to 14 volts or more. We tried using the DAC1220 on two SS50 bus machines and found that it worked as well in them as it did in SS50C systems. Upon checking, we found that one machine had \pm 13.5 volts on its 12 volt bus lines and the other had 14 volts. So you may be able to use the DAC1220 in an older unmodified machine the Meta Lab does not guarantee that you can. So you can try it and if it does not work properly, you can upgrade the power supply as described in the article by Russell Gorr in the June 1980 issue of 68 Micro Journal.

We feel the Meta Lab DAC1220 is a professional quality product. It is extremely versatile and exceeding easy to use. The priced out of the range of most hobbyists; perhaps Meta Lab could make the DAC1220 available as a bare board or kit of parts to allow more hobbyists access to this fine product. The DAC1220 is very deserving of a AAA rating.

Mickey Ferguson POB 87 Kingston Spgs., TN 37082

TSC

ASSEMBLER PRINT UTILITY

One day while printing an assembler listing with TSC's 6809 assembler, it occurred to me that if the output was condensed, no lines would be wrapped by the EPSON MX80 printer, and if the line spacing could be reduced to 8 lines per inch, paper would be saved, and the whole listing would be more compact and easier to read. (As long as you have good eyes, good eyeglasses, and/or a new printer ribbon.)

That started me on a project to make a simple way to do all this, and the program that follows is the result. Along the way, another desire crept into the plan--indentation of the listing to provide room to punch holes for a 3-ring binder.

At first, I attempted to use the EXEC function to accomplish what I wanted, but it became apparent that a little more creativity was required. But don't let me take you through the development process; it isn't worth the words.

The resultant program, printed below, does the following: the EPSON MX80 printer (with GRAFTRAX OR GRAFTRAX+) is set to 8 lines per inch, 88 lines per page (for 11 inch paper), and condensed mode (132 characters per line). Also, the indentation mentioned above was included to move the listing to the right by 8 characters to leave a wide left margin.

The program listing is fairly well commented, so not too much in addition needs to be said. But some general comments are in order. First, the program uses FLEX utility commands to execute portions of the total function. This is accomplished by inserting a command line, as is normally entered from the keyboard, directly from the program. FLEX is then called to execute this command line, and then to return to the program which called it. The functions executed this way are "P", which loads the PRINT.SYS driver, and "TTYSET WD=0", which sets the line width to infinity. Finally, the "GET 0.ASMB.CMD" loads the assembler into memory. Before this command line is transferred to the FLEX line buffer, the current value of the line buffer needs to be saved, since it includes the filename to be assembled and the assembler options. This is what the code does at SAVLP. The code around STRCMD puts our new command line in the FLEX line buffer. JSR DOCMND performs the functions requested by the command line in the line buffer, and then returns. The code around PRTLP sends the printer the control commands required.

The assembler is modified at \$0018 to set 80 lines per page. The TTYSET pause option is turned off if on, and the address of the print output character routine is placed into OUTCHR in FLEX. Note that this routine is in this program in order to add the indentation desired at the start of each line.

After all this malarky is complete, the original command line is returned to the line buffer, along with the proper setting of the FLEX line buffer pointer, all this around RESTOR. A JMP 0 instruction transfers control to the assembler.

The PRINT routine is called when the assembler outputs a character to the printer. This is the routine which adds the wide left margin. This overly complex routine is required to keep track of lines which contain only line feeds and carriage returns. It was developed empirically and thus has

no theoretical basis. But it seems to work.

After you get this small utility on your system disk as a file called PASMB.COM, you can then key in the following to get a compact listing: PASMB <assembler source file> <assembler options>.

I start each assembler source file with the following three statements:

```
NAM <program title to appear on each page>
OPT PAG
PAG
```

In addition, I include 'PI' as the last assembler option in the command line. This combination makes Page 1 the first page printed, starting with the source following the PAG instruction shown above. The program title and the date is printed at the top of each page.

This routine loads into the upper area of the FLEX utility command space. This is a tricky, dangerous technique since during the course of execution, other utility commands get loaded into the lower part of the same space (C100-C6FF). But, again, it seems to work.

Also, since this program mucks around with TTYSET parameters, (PS & WD), and doesn't restore them, they may end up different than their original values.

Two final points: First, it is possible to play with the FLEX command line and make it do what you want from a program; and second, this little program does produce an assembler listing that is more compact and has almost no line length restriction.

Walt Cole 5868 Pentz Way San Jose, CA 95123

PRINT ASSEMBLED LISTING

8-28-83 TSC ASSEMBLER PAGE 1

```
* PASMB PROGRAM      28 AUGUST 1983
*
* This program sets up the assembler and an EPSON M100
* printer to assemble a program and print the output
* listing in a condensed format. The output will be at
* 132 chars/line, so that carefully entered comments
* will not be truncated nor carried to the next line.
* The program will set the printer to 8 lines per inch,
* to produce an 88-line form length.
* The assembler itself is patched after loading so that
* the number of lines per page is increased from 36 to 88.
* OUTCHR in FLEX is changed to code to PRINT in this
* program each time a character is printed.
* The PRINT routine indents each output line by 8 spaces,
* leaving room to punch holes for a binder.

* EQUATES

C000  LINDXUF  EQU  0C000  FLEX line buffer.
C014  LINDPTR  EQU  0C014  FLEX line buffer pointer.
C020  REHEND  EQU  0C020  Recovery end.
C0E4  POUT  EQU  0C0E4  Printer output CMMR routine.
C0B3  MARKS  EQU  0C0B3  FLEX macro start entry point.
C06F  OUTCHR  EQU  0C06F  Output CMMR to terminal.
C243  DOCMND  EQU  0C243  Call FLEX DOS as subroutine.

*
C588
C588  START  ORG  0C588  Upper area of FLEX utility
                        command space.

* Save the FLEX line buffer for later use when the
* assembler gets control.

C500  BE  C000          LDX  0LINDXUF  FLEX line buffer addr -> I.
C508  1PBE  C672        LDY  0LINDPTR  Save area addr -> Y.
C50F  A6  00           LDA  #I+      Line buffer char -> A.
C5C1  A7  A6           STA  #Y+      A -> Save area.
C5C3  8C  C100         CMPI  0LINDUF+128
C5C6  2D  F7           BLS  SAVLP     Branch until 128 chars saved.

* Then save the FLEX line buffer pointer for later use.

C5C8  BE  C014          LDX  LINDPTR
C5CB  BF  C6F2          STX  PTRSAV   Save line buffer pointer.

* Store our new command line into the FLEX line buffer.

C5CE  BE  C000          LDX  0LINDXUF  FLEX line buffer addr -> I.
```

C5D1 10BE C64E	STRCD	LDI	0CHDSTR	Addr of new command string -> Y.
C5D2 A6 00		LDA	0,Y	New cmd string char -> A.
C5D7 A7 00		STA	0,Y	A -> FLEI line buffer.
C5D9 10BC C66B		EMPTY	ENDSTR	
C5D0 29 F6		BLT	STRCD	Branch until end of cmd string.
C5D6 0E C660		LDI	0LINBUF	Set line buffer pointer to
C5E2 BF C614		BTI	LBPTTR	beginning of command string.
C5E5 0D C64B		JSR	0OCMD	Let FLEI do the hard work.
C5E8 C1 00		CMP	00	Check for FMS error.
C5EA 26 5F		DNB	ENDOR	

* Set up printer for 132 chars/line, 0 lines/inch and
* 00 lines/page.

C5EC 0E C66B	LDI	0PRCTL	Printer ctrl string addr -> I.
C5EF C6 07	LDB	07	7 ctrl chars to send.
C5F1 A6 00	LDA	0,Y	Printer ctrl char -> A.
C5F3 0D CCE4	JBR	0OUT	Send ctrl char to printer.
C5F6 5A 00	DECD	0	Decrement counter.
C5F7 26 F8	BNL	PRCTL	Branch until 7 chars sent.

* Now the assembler is loaded, PRINT.SYS is loaded, and
* the printer is set for compressed print, 0 lines/inch,
* and 00 lines/page.

* Next, overlay assembler for 00 output lines per page.

C5F9 06 50	LDA	1450	00 lines per page.
C5FB 97 10	STA	10010	Store in assembler.
C5FD 7F CC09	CLR	1CC09	Turn off pause if on.

* Store address of print char routine into FLEI.
* Note that PRINT is a routine in this program!

C600 0E C621	LDI	0PRINT	Print char addr see below.
C603 BF C610	STI	0OUTCH	Stuff in FLEI.

* Restore FLEI line buffer from save area.

C606 0E C680	LDI	0LINBUF	FLEI line buffer addr -> I.
C609 100E C672	LDI	0BUFSAV	Save area addr -> Y.
C60B A6 00	LDA	0,Y	Get char from save area.
C60F A7 00	STA	0,Y	Store char into line buffer.
C611 0C C100	CMP	0LINBUF+120	
C614 20 F7	BLT	RESTOR	Branch until 120 bytes restored.

* Restore FLEI line buffer pointer.

C616 0E C6F2	LDI	0PTRSAV	
C619 BF CC14	STI	LBPTTR	
C61C 7F C6F4	CLR	FLAG	Used by PRINT routine later.
C61F 0E 00	JMP	0	Jump to start of Assembler.

* The PRINT routine is executed each time the assembler
* outputs a character to the printer. 0 spaces are output
* to the printer if a new line is being started. This
* causes a wide enough left margin in the printed listing
* to punch holes for a binder.

C621 34 10	PRINT	PSHS	I, B, A	Save I, B, and A regs.
C623 01 0A		CMPA	010A	
C625 27 1F		BEB	SETFLG	Branch if char = line feed.
C627 10 C6F4		TEST	FLAG	
C62A 27 15		BEB	NORMAN	Branch if no spaces needed.
C62C 01 0D		CMPA	010D	
C62E 27 11		BEB	NORMAN	Branch if no spaces needed.
C630 7F C6F4		CLR	FLAG	
C633 C6 00		LDB	00	Indent 0 spaces.
C635 06 20	SPACLP	LDA	0120	
C637 34 04		PSHS	0	
C639 0D CCE4		JSR	0OUT	Go print a space.
C63C 35 04		PULS	0	
C63E 36 00		DECD	0	
C63F 26 F4		BNL	SPACLP	Branch until 0 spaces printed.
C641 35 10		CMPA	I, B, A	
C643 7E CCE4		JMP	0OUT	Print char from assembler.
C646 7C C6F4	SETFLG	INC	FLAG	
C649 20 F6		BRA	NORMAL	

C640 7E C005	ENDOR	JMP	00000
C64E 50 20 34 34	ENDSTR	FCC	'P TTYSET WD=0:'
			Load PRINT.SYS and
			allow long lines.
C65C 17 43 34 20		FCC	'GET 0.ASYM.END'
C66A 00		FCC	000
			End of cmd string.
	ENDSTR	EDU	0

C660 07 0F	PRCTL	FCB	107,40F	Condensed print mode.
C663 10 30		FCB	010,430	0 lines per inch.
C66F 10 43 50		FCB	010,443,030	00 lines per page.

C672	BUFSAV	RMB	128	Line buffer save area.
C673	PTRSAV	RMB	2	
C674	FLAG	RMB	1	
		END	START	

0 ERROR(S) DETECTED

SUPPORT YOUR ADVERTISERS

XDMS

XDMS DATA MANAGEMENT SYSTEM

The XDMS Data Management System is a complete system which allows the user to define and update structured data files, and subsequently produce report, inquiry and processed file output with ease. The system supports sequential, hierarchical sequential and random file structures, which may be logically interrelated by common keys. This relational aspect permits the viewing of a collection of discrete files as a "database" without the complexity normally found in database management systems.

XDMS is an extremely powerful system which supports such processing functions as record selection, selection via a reference file, merging of data among files, reformatting of files, sorting, aggregation of records, calculations, special processing of file subsets, merging of text and data and customized forms generation. This functionality is available when needed, and may be employed by the user to suit a particular application, via an English-like command language.

Typical examples of use are schedules, accounting data, checkbooks, addresses, inventory, records retention, market trends, mailing lists and most other application requiring the structuring of data. The XDMS system, when properly applied, can displace the need for many specialized programs and, thus, reduce long term software costs. The XDMS system consists of the following facilities:

The DEFINE facility allows the user to define the format and content of a DMS file. Field name, format and size, and groups of fields are identified to the system and the definition becomes a permanent part of the data file. This enables later reference and titling of the data by name. The file structure may be hierarchical, which reduces disk space required, and up to 24 fields or groups may be defined per file. Alphabetic, numeric, coded, integer, decimal, scalar date and hexadecimal fieldtypes are supported, with all but alphanumeric being compressed internally. This permits larger files to exist on limited disk space. Since the definition resides on the file itself, XDMS data files may be manipulated by standard utility commands.

The UPDATE facility allows the user to input, review, change and delete data file information. These editing functions are executed via the system terminal on an interactive basis. UPDATE may also be used to input data files from formatted text files. Single character commands facilitate execution of desired functions with minimum effort. UPDATE also features conveniences such as field copying, ability to string commands and user macro definition.

The GENERATE facility provides powerful processing capability within the XDMS system. English-like instructions are utilized to produce printed reports, file listings, processed data file output and formatted terminal output. GENERATE may be applied in selection, inquiry, analysis and processing applications, eliminating the need for user written programs. The instructions may be stored in a control file, and the desired processing invoked by entry of two words.

GENERATE functions include match and range selection of input records on up to 15 fields, record selection based upon a reference file which may be preselected, formatting of selected fields for output, merging of data from other files via multiple keys, definition of new output fields and groups, sorting of records on up to 15 keys, field array generation, record aggregation, totaling of selected fields, insertion of related file records as heading data, inclusion of "link" file data which forms an indirect link among files, record and field merge with user specified text, and formatting of output on the terminal, printer or disk. Command files may be easily created via a text editor, and may contain symbolic parameters, for which the user is prompted at execution time. This permits inquiry, and other applications which require variable execution. Data file output may be re-read within the same execution, thereby permitting complex processing functions if required.

The extended forms feature of GENERATE defines an output "page" which consists of optional insert, link and main file tabular data enclosed by user defined heading, form, body and footing text. Further, field data may be inserted in with the text if desired. This permits complex forms such as invoices and custom contracts or form letters to be produced based upon data file content.

The XDMS Data Management System employs a resident module known as the XDMS nucleus. This module contains approximately 200 service functions which are employed by XDMS applications and includes data manipulation, arithmetic, file and field management, virtual storage management and disk, terminal and printer I/O functions. Many of these functions emulate those found in large scale computers. This configuration affords maximum control, flexibility and standardization, and substantially reduces the size of application programs which run under XDMS.

XDMS employs a virtual memory technique to increase the effective memory available for user data. This methodology utilizes an entire disk drive as a direct access "paging device" to swap in and out sections of memory as they are required. The result of this is the ability to directly access files as if they were in memory; a method which is extremely fast when compared to conventional disk access methods. The page disk is requested as needed allowing small applications to use both drives.

Included with the XDMS system is a VMGEN utility. This command is used to format the pagind disk used by the system. XDMS searches for this disk on a memory overflow condition and requests that the user insert the disk if it is not found.

The XDMS Data Management System described here is intended as a basic system upon which other specialized software packages may be built. The nucleus itself, provides adequate functionality to support almost any application. The other facilities described provide the necessary

DID YOU KNOW?

... that the \$280 ELEKTRA computer cabinet is the largest SS-50 cabinet available?

No more cramped space when using an ELEKTRA computer cabinet. With the ELEKTRA universal motherboard, you have a 1-15/16" clearance behind and in front of the motherboard as well as 1-3/16" clearance to both the left and right of the motherboard for oversized boards. Also, all I/O and memory slots are spaced a full inch apart center to center for ease in removing and installing boards. Need more room? Shift the motherboard off center! (Actually, no one has reported doing that yet.)

... that the \$278 ELEKTRA Universal Super Floppy Controller has 13 option areas (24 minijumps) making the ELEKTRA Super Floppy Controller the perfect upgrade for your current system.

... that the ELEKTRA Winchester Systems have automatic error correction up to 11 bit burst errors. Buy the best Winchester systems available! Mixed systems also supported. Now you can add a Winchester system even though your floppy disk system is based on a 30 pin floppy controller.

... that the Engineer's Fun Box at only \$228 is the perfect box for that second system, control system, or whatever, where you're using a standard size motherboard.

... that the ELEKTRA HD-SW cabinet is perfect for holding 4 half height 8-1/4" floppies.

... that you can run the 68008 on your SS-50 bus.

The HELIX 68008 CPU board with CP/M-68K is waiting for your SS-50 bus.

Need FLEX, UniFLEX, OS-9 Level I, or OS-9 Level II? We have a system for you!

Spending too much money? We have the quality hardware needed to service your needs!

Phone **AAA Chicago Computer Center**. Technical consultation available most weekdays from 4 PM to 6 PM CST. (312) 459-0450.
120 Chestnut Lane — Wheeling, IL 60090

See our catalog and ordering information on the next page to your right.



ANOTHER WINNER! AAA Chicago Computer Center is proud to announce the addition of the HELIX line of computer products. The HELIX line, together with SWTPC, SSB, and ELEKTRA, permits us to offer you the greatest selection of quality components to handle all of your needs.

84K 6809 Computer	\$2395.00	256K 68000 Computer	\$3695.00
256K 6809 Computer	\$2895.00	512K 68000 Computer	\$4195.00
Other systems available			
20 Megabyte 5" add on Winchester System			\$2595.00
64K CMOS Static memory board with battery backup			\$95.00
DMA 5" and 8" Floppy Controller with built-in Winchester controller I/O			\$695.00
DMA 5" and 8" Floppy Controller	\$495.00	6809 CPU Board	\$495.00
68008 board for SS-50	\$595.00	CP/M-68K	\$350.00

Need FLEX, UniFLEX, OS-9 Level I, or OS-9 Level II? We have a system for you!

ELEKTRA COMPUTER CABINET THE LARGEST SS-50 COMPUTER CABINET AVAILABLE! Made of heavyweight 0.090" thick aluminum. Interior is 18 1/2" wide by 21 7/8" deep by 8 3/4" high. Heavy duty A.C. line cord. A.C. fuse holder. EMI filter. Fan with filter. Back panel has 10 cutouts for D-type data connectors. Front panel has on/off power switch, 2 illuminated push button switches (Reset and NMI/Abort), and two cutouts for 5 1/4" disk drives. \$250.00

Filter Plate for 5 1/4" drive opening \$10.00 Fan Filter \$10.00

POWER SUPPLY Highest quality linear power supply CONSERVATIVELY rated at 15A @ 5V, 3A @ 16V, 3A @ 10V. 3 primary inputs for light, rated, and heavy loading. 220V Version: \$200.00 110V Version: \$175.00

DISK REGULATOR BOARD WITH CABLES Standard version for 2 floppy drives \$50.00 Heavy duty version for 1 Winchester drive and 1 floppy drive \$75.00

ENGINEER'S 'FUN BOX' BY ELEKTRA Computer cabinet with high quality 10 amp power supply, EMI filter, fan. Large enough to hold the standard size SS-50 type motherboard. \$225.00

ELEKTRA UNIVERSAL SS-50/SS-50C MOTHERBOARD Heavyweight 0.125" thick, 18" long by 9" wide. 11 memory (50 pin) slots, 8 I/O (30 pin) slots. Complete address decoding and selection for I/O slots. Choice of 4, 8, or 16 addresses per I/O slot. 1" spacing between all memory and I/O slots. On board baud rate generator with low and high ranges providing jumper selectable rates of 75 through 38,400 for each of the five baud rate lines, slow device circuitry permitting 1 Mhz 30 pin disk controllers to run with 2MHz 50 pin CPU boards.

Mounting hardware \$5.00 Bareboard w/ documentation \$80.00 Kit w/ gold connectors \$320.00 Assembled w/ gold connectors \$380.00 Kit w/ tin connectors \$240.00 Assembled w/ tin connectors \$300.00

ELEKTRA CHASSIS Includes cabinet, 110V power supply, power supply cables, standard disk regulator board with power cables, motherboard with gold square pin connectors, assembled and tested. \$850.00

ELEKTRA CPU 8/9 Use either the 6802 or 6809 (to run 6800 software) or 6809. Has provision for up to 32768 E-proms, 1K scratchpad, M6840 triple timer, and an optional baud rate generator providing baud rates from 110 through 38,400 baud in two user selectable ranges. Versions of OS-9 level I are available. Bareboard: \$50.00 Kit: \$225.00 Assembled: \$275.00 Optional Baud Rate Generator: \$25.00

ELEKTRA OPS DUAL PORT SERIAL CARD Fits the standard 30 pin SS-50 bus I/O slot. Can be configured for 4 or 16 addresses per port. RTS, CTS, DTR, DCD, IRQ, FIRQ/NMI, and baud rate can be appropriately implemented for each port. Bareboard: \$20.00 Kit: \$60.00 Assembled: \$80.00 Cable with mounting hardware (two needed per board) Each \$25.00

ELEKTRA OPP DUAL PORT PARALLEL CARD Fits the standard 30 pin SS-50 bus I/O slot. Can be configured for 4 or 16 addresses per I/O slot. The direction of the TTL buffers can be controlled by either on board jumper connectors or by a signal from the peripherals. The interrupt request line for each port may be individually jumpered to either the IRQ or FIRQ/NMI bus line. Bareboard: \$20.00 Kit: \$60.00 Assembled: \$80.00 Cable with mounting hardware (two needed per board) Each \$25.00

ELEKTRA 84K STATIC RAM/ROM MEMORY BOARDS with gold connectors (tin available) Assembled and tested. With 56K RAM \$269.00 With 64K RAM \$299.00

ELEKTRA UNIVERSAL SUPER FLOPPY CONTROLLER THE BEST 30 PIN FLOPPY DISK CONTROLLER THAT YOU CAN BUY! Controls up to four 5 1/4" drives and four 8" drives for a total of eight system drives. (FLEX system limit is four drives.) Single density or double density. 1MHz or 2MHz, 6800 or 6809. (Double density, 8" must be at 2MHz, all other combinations of performance are possible.) Analog Phase locked loop data separator with separate adjustments for 5" and 8" drives. Analog write precompensation circuit with separate adjustments for 5" and 8" drives. Designed to meet the data hold requirements of Western Digital floppy controller IC. Assembled and tested. \$275.00 Disk with drivers and formatting utilities (Specify 6800/9, FLEX/OS-9) 30.00

ELEKTRA WINCHESTER SYSTEMS THE BEST WINCHESTER SYSTEMS THAT YOU CAN BUY! Has automatic error detection and CORRECTION of up to 11 bit burst errors. SS-50 bus extended addressing capabilities, DMA, on board sector buffer, drivers included for 6809 FLEX or OS-9. Specify whose version of FLEX that you are using. Drivers for FLEX2 are available for an additional \$100.00. Price includes host interface, controller, driver(s), and cables.

12 Megabyte single drive sys. \$2295.00 24 Megabyte dual drive sys. \$3595.00 19 Megabyte single drive sys. \$2595.00 38 Megabyte dual drive sys. \$4695.00 (19 Megabyte drives are the largest that can be supported by FLEX)

ELEKTRA HD-5 Cabinet for dual 5 1/4" floppy drives with power supply, fan, fuse, power switch, and power cables to drives. \$150.00

ELEKTRA HD 5W As above, but with EMI filter, fan, and heavy duty power supply. Powers 1 floppy and 1 Winchester. \$199.00

5" ribbon cable for dual 5 1/4" disk drives. 40.00

ELEKTRA HD-8 Dual drive cabinet, EMI filter, fan, power supply, and power supply cables for 8" drives. 350.00

8" ribbon cable for dual 8" disk drives. 45.00

ELEKTRA 30 PIN PROTOTYPING BOARD 20.00

ELEKTRA 50 PIN PROTOTYPING BOARD 40.00

GOLD 10 PIN CONNECTORS (Specify male with square pins or female) 1.50

TIN 10 PIN CONNECTORS (Specify male with square pins or female) .50

ELEKTRA is a trademark of AAA Chicago Computer Center. FLEX and UniFLEX are trademarks of Technical Systems Consultants, Inc. HELIX is a trademark of Hazelwood Computer Systems.

OS-9 and BASIC are trademarks of Microware Systems Corp.

Dealer for ELEKTRA, HELIX, SSB, SWTPC, Microware Systems Corp., and Technical Systems Consultants, Inc.

Prices and inventory are subject to change without advance notice. These ads are our catalog.

AAA CHICAGO COMPUTER CENTER
120 CHESTNUT LANE • WHEELING, IL 60090 (312) 459-0450
Technical consultation available 4 PM to 6 PM most weekdays. Closed evenings and weekends.

TERMS Minimum order \$20.00. Shipping and handling estimates within the Continental U.S., add 3% (MINIMUM \$2.50). Illinois residents add 6% sales tax. We will refund your overestimated shipping and handling charges. Foreign shipping and handling, add 10% (MINIMUM \$10.00). Foreign orders must be prepaid in U.S. dollars. Checks must be drawn on a U.S. (or Canadian) bank. Heavy foreign items will be shipped air freight collect. Please phone between 4 PM and 6 PM weekdays if questions arise regarding shipping fees. Master Charge, Visa, and American Express honored.

Our apology. We are not staffed to answer technical inquiries through the mail. Please phone for technical help during the hours indicated above. The too frequent changing of our inventory and prices makes it uneconomical to publish a catalog. Our ads are intended to serve that purpose. Prices and inventory are subject to change without advance notice.

ELEKTRA SOFTWARE (All of our software is copyrighted and all rights are reserved. Source is either supplied or optionally available at extra cost so that the purchaser can modify our programs for his own use. Licensing, however, is required for commercial resale.)

SUPER MODEM PROGRAM Single character commands. No inter-ups required. Transmit manually or transmit disk files (text) of any length to distant computer. Receive and save disk files (text) on local disk system. X-on/X-off supported. Tested for full duplex at speeds up to 6800 baud. Half duplex option. Echo option. Replaces CR with CR/LF (user option). Slow disk file transmit option.

Please specify 6800 or 6809, SSB or FLEX™, 5" or 8" instruction Manual and disk with both source and object code \$75.00

STANDARD MODEM PROGRAM Same as Super Modem Program above but without ECHO option, CR/LF for CR option, slow disk file transmit option, nor X-on/X-off option. Specify 6800 or 6809. Manual with instructions, source listing, and flow chart 10.00

OS-9 Modem Program 100.00

ORDER — WRITE UP COMPUTER PROGRAM Screen oriented write up form with cursor editing, disk save and load, printer command using easily available universal print-out forms. Phone for more details. Available for 6809 FLEX. \$100.00

ALL IN ONE Editor — Text Processor — Mailing Labels — Mailing Lists — Multi-Form Letters. Use any CRT terminal and printer — Best Package For The Money Anywhere!

Specify 6800 or 6809, SSB or FLEX™, 5" or 8" 50.00
Printed source listing is available for an additional 35.00
All-In-One, Write n spell, and Spell n Fix package 250.00

Software by Technical Systems Consultants, Inc.

	UniFLEX™ w/1 yr. mnt.	FLEX™
DOS (includes Editor and Assembler)	550.00	150.00
Editor or Assembler		50.00
68000 Cross Assembler on 6809	300.00	250.00
6800 FLEX™ Utilities		100.00
Text Processor or Sort/Merge Package or 6809 FLEX™ Utilities	150.00	75.00
Extended Basic	200.00	100.00
Basic Precompiler for Extended Basic	150.00	50.00
Pascal	300.00	200.00
Debug Package or Diagnostic Package	75.00	75.00
6809 Relocating Assembler & Linking Loader	175.00	150.00
Fortran I (With Relocating Assembler & Linking Loader)	450.00	
Cobol	750.00	500.00

Software by Microware Systems Corp.	Run-Time Package	Update	Source	Manual Only	Object w/Man.
OS-9™ Level One Operating System		75.00	400.00	40.00	200.00
OS-9™ Level Two Operating System		75.00	N/A	40.00	500.00
BASIC®	100.00	75.00	N/A	25.00	250.00
OS-9™ Macro Text Editor			300.00	15.00	125.00
OS-9™ Interactive Assembler			300.00	10.00	125.00
OS-9™ Interactive Debugger (Disk version)			100.00	10.00	50.00
C S Cobol Compiler	400.00	50.00	N/A	80.00	900.00
Pascal Compiler	100.00	100.00	N/A	40.00	400.00
C Compiles		100.00	N/A	40.00	400.00
Microware yearly support service \$200.00 for OS-9 Level 2)					

Special Software
MICROBUG (2K, 6809 Baby HUMBUG by Peter Stark) 30.00
4K 6809 HUMBUG 75.00
4K 6800 HUMBUG (RAM needed at \$4000 and \$D000) 65.00
2K 6800 HUMBUG (With cassette LOAD and PLUNCH) 40.00
2K 6800 HUMBUG (Extra commands instead of cassette software) 40.00
Other HUMBUG versions including video versions are available.

Spell n Fix by Peter Stark 178.50
Write n Spell by Peter Stark 75.11
All-In-One, Spell n Fix, and Write n Spell package 250.00
Dynamic Disassembler 60.00
SUPER SLEUTH Disassembler System (\$101.00 for OS-9 version) 99.00

SPECIALS

- U.S. Robotics 300/1200 baud auto dial/auto answer modem 499.00
- Same as above but without self test and diagnostics 399.00
- U.S. Robotics 1200 baud direct connect auto answer modem 349.00
- Hazeltine 1420 CRT terminal (new) 375.00
- SSB BFD Floppy Disk Controller (Version 3) Run FLEX or SSB DOS 100.00
- SWTPC 4K Memory \$15.00 MP-B \$40.00 MP-C 10.00
- MP-S Other HUMBUG versions including video versions are available. 25.00
- High speed tape reader 50.00 AC-30 \$40.00 MP-A2 125.00
- 300 Baud acoustic modem 129.00 F & D PMB-1 Video Assembled 60.00
- 10 ELEKTRA DD 8" Disks 35.00 10 5" DD Disks in hard box 25.00
- 1 J180 Printer w/ lower case and full vertical forms control 1200.00
- Microtime II Calendar and Clock Board (Assembled) 60.00

SD/DD DISK DRIVES	2 heads	1 head	2 heads
30 day guarantee	CDC	MP1	MP1
5 1/4" 40 tracks	300.00	250.00	325.00
5 1/4" 80 tracks	375.00	325.00	400.00
MP1 or CDC Service Manual (Specify 40 or 80 track)			25.00
8" 77 tracks DS/DD	Qume DT-8 \$550.00	Remax (Special)	350.00

GIMIX CLEARANCE SALE

LIST	OUR PRICE	LIST	OUR PRICE
es 6809 Plus CPU Bd.	578.05	6800 CPU board	4.03 100.00
Cable (Ser or Par I/O)	24.95	Baud rate gen. board	88.93 65.00
Motherboard	200.00	8 Port Serial I/O Bd.	318.46 250.00
Double disk reg. card	68.22	428 control w/GMX Fies	328.28 270.00
32K memory board	175.00	56K memory board	425.00
64K memory board	478.67	Single prt ser. 1 cable	113.38 90.00
80 x 24 Video Boards	388.78	Dual prt ser. 2 cables	138.32 110.00
64 x 16 Video Boards	198.71	4K PPD PROM Bd. and burner	100.00
16K Mem Bds. w/cntrl reg.	145.00		
93L422 RAM chips (2 needed for GIMIX DAT)		each	20.00

SWTPC

6809 SW PC FLEX™ and manual			35.00
DC-4 Disk Controller (SS/DS, SD/DD, 5 1/4")			230.00
MP-S2 Dual Port Serial	120.00	MP-L2 Dual Port Parallel	120.00
MP-N Calculator Board (kit)	54.95	MP-N (assembled)	92.00
MP-A 2716 Eprom Programmer	114.50	MP-08 2MHz 6809 CPU Board	295.00

Emulex Signal Broadcasting

SSB version of FLEX™ (without Editor and Assembler)	150.00
LMB-1A Motherboard	399.00
SCB-69 6809 CPU Board	399.00
PAR-1 Dual Port Parallel Board	89.00
SER-2 Dual Port Serial Board with 2 Cables	129.00

WARNING AAA Chicago Computer Center does not provide repair or diagnostic service for customer assembled kits. AAA Chicago Computer Center does warranty and maintain service for our assembled boards. The customer should carefully take into consideration the small differential separating out kit and assembled prices when making his choice of purchase.

We have introduced our line of computer equipment with the purpose of offering the highest quality of components possible at affordable prices. These products are intended for OEM applications where it is the responsibility of the purchaser to integrate these components with suitable memory, disk controllers, drives, and software along with I/O terminals to form working computer systems.

The XDM5 system is available on 5" or 8" disk for only \$179.95.

8088 MICRO MAY
Anywhere-on-Earth, 08700
(333)-333-1212

*** INVOICE ***

SHIP TO

Your Valued Customer, Inc.
40 Small Business Circle
Burgersville, U.S.A. 14001

Order Information

0401X
4EFO 0401X-- SUK4--- ACC10 SHIP 0110 SHIP--- TERM-----
0700 A10003 SUFL 003 1002 0073 1002 U.F.S. M01 30 0450

Your order for the following items is enclosed. Any remaining items indicated will be shipped as soon as availability permits.

PURCHASE ITEMS		QUANTITY		PRICE		EXTENDED
LINE	UNIT/ITEM	ORD	SHR	RE	EACH	
1	1/2inchx4feet, 12" x 4", Red	24	24		3.75	90.96
2	1/2inchx4feet Crosscut Saw	1	1		6.99	6.99
3	1/2inchx4feet Paint, Green	2	2		1.79	3.58
4	1/2inchx4feet, 4" x 4" x 3" H	48	48		5.59	268.32
5	1/2inchx4feet, 1/2" x 4" x 3" H	48	48		1.79	37.92
6	1/2inchx4feet, 1/2" x 4" x 3" H	48	48		1.99	95.52
7	1/2inchx4feet, 1/2" x 4" x 3" H	2	2		3.99	7.98
8	1/2inchx4feet, 1/2" x 4" x 3" H	48	48		.59	28.32
9	1/2inchx4feet, 1/2" x 4" x 3" H	48	48		.89	42.72
10	1/2inchx4feet, 1/2" x 4" x 3" H	48	48		.88	33.12
11	1/2inchx4feet, 1/2" x 4" x 3" H	1	1		9.75	9.75
12	1/2inchx4feet, 1/2" x 4" x 3" H	48	48		1.29	61.92
						879.96

This invoice illustrates some of the advanced features offered by the SAGE Data Management System. The general report model used provides for: 1) optional inclusion of 1) a header block, 2) 1) User-defined text, 3) Inserted file data, 4) User form text, 5) User file data, 6) User body text, 7) Main file data, 8) Totals and 9) User footing text, used here to describe the form. Additionally, field data may be interleaved with any of the four text blocks, permitting business letters, bills, contracts and other user Customized forms.

XDMS DATA MANAGEMENT SYSTEM

VERSION NOTES

XDMS VERSION 1.0 NOTES

This section is provided to document changes to the XOMS system. In general, new version numbers are assigned when changes are made to the XOMS nucleus, which implies re-assembly of all associated software, or when sufficient functional enhancements are applied to the system.

Version 1.0 is the initial issue of the XDMS Data Management System. XDMS is an "extended" DMS based upon the DMS2/VM system produced for and marketed by Mestcheater Applied Business Systems, Inc.

The XOMS system has been designed to be upward compatible with DMS2/VM, however, the reverse is not necessarily true. The following items highlight the differences for those who are migrating from DMS2/VM to the XOMS system.

t. The entire nucleus has been rewritten to add functionality, and to condense the code where possible. IDMS makes extensive use of direct page registers, counters and temporary storage areas. As a result, many instructions are of the "direct" mode and occupy 2-3 bytes instead of 3-3. This permits approximately 280 functional subroutines to exist in the 8K allocated to the nucleus, and also speeds up processing since direct mode instructions use less MPU cycles.

2. XDSM user data files have a .OMS extent while DMS2/VM files have a .ESD extent. DMS2/VM design provided for "Entry Sequence Data (ESD)" and "Key Sequence Data (ESD)" - factors the latter was never used, which is a common mistake. Hierarchical files are stored in order up to 128 bytes. XDSM defines "Data Management System (OMS)" files which may be hierarchical, sequential or random with records up to 255 bytes. Since the XDSM definition encompasses the DMS2/VM definition, existing ESD files may be used with XDSM, by renaming the extent to "OMS". Caution is advised in the reverse order, since OMS files may exceed limitations imposed by the DMS2/VM design. XDSM uses .DMS extents for backup files.

3. Considerations of users with narrow screens and/or color computers have been employed. Interactive prompts and headings are output in capital letters; all commands are input as capital letters and most interactive input and output has been arranged to fit within a 32 character line to avoid screen wrap. This does not effect users with larger screens.

4. Management of ASB and TTYSET variables is assumed by XDMS. This permits better control of disk accessing and report formatting. It is also part of a long term migration strategy to other operating systems. TTYSET variables should be left defaulted when using XDMS to avoid conflict.

5. All forms generation under XDMS is performed by a GENERATE facility.

Essentially, GENERATE replicates the DMS2:VM GENER and FORMAT functions and incorporates additional forms and processing capabilities. While similar, the control language used has been revised to provide for the additional functionality and to be more user friendly and English-like. DMS2:VM users may need to modify existing .CTL files with respect to the XDMS syntax.

XDRMS DATA MANAGEMENT SYSTEM

GENERATE FACILITY

SYNTAX SUMMARY:

The following summarizes GENERATE statements and may be used as a "checklist" of functions to be performed. The syntax is non-procedural so that statements may be ordered as desired, provided that file and field references can be resolved.

```

DRIVE (o))
[TITLE (test lines...) (new line)] (1 per run)
[MARGIN (val1,val2)]
[PAGE (val1,val2)]
[FILE (o, filename) (1 per run)]
[INCLUDE (o, filename)... (Up to 12 per run)]
[SELECT (fld1) WHEN (f1,f2,fld2) (=(<f1>|o) (value) (1 per run)]
[COUNT]
[IFOR (fld1) (=(<f1>|o) (value)...(Up to 15 per run)]
[LIST(COPT)]
[NAME (filename)]
[RANDOM]
[PRINTISAVE (fld1,fld2 [AND] fld3...[AS fld4]) (new line)]
[CACHE (fld1)]
[GROUP (name)]
[DEFINE (name,type,fmt) [FILL (value) (new line)]
[MERGE (o, f1,f2,fld2) USING (fld1)]
[ARRAY (fld1) FOR (fld2) (=(<f1>|o) (val1) [AND] (val2)... (new line)]
[EXCLUDE]
[DUPLICATE]
[FOLD (grp1*)]
[TABSET (val)]
[SEPARATE (char)]
[SUM(SORT) BY (fld1*) [BY (fld2*)]...]
[CALC (fld1=REGx|val) [(o1=|o1/ fld2o1REGx|val)]...[opr] (ofld3=|REGx)]
[ON (fld1 o)]
    [LINK (o, f1,fld1)]
    [INSERT (g, f1)]
    [SUBTOTAL (fld1o),fld2o)... (new line)]
    [SKIP (val)]
    [EJECT]
    [CLEAR]
[TOTAL (fld1*)[(f1,fld2*)... (new line)]
[NO-HEADING]
[NO-INSERT-LABEL]
[NO-LINK-LABEL]
[NO-DATA-LABEL]
[TYPE]
[TEXT]
[HEADING:FORM:BODY:FOOTING (new line)
    (test lines...)]
ENDX (new line)]
[QUIT:ENDIRUN] (Note: ENDIRUN requires a FILE statement)

```

* Indicates that the referenced field or group must exist in the output file. This is established by a previous LIST, PRINT, SAVE or other statement which includes the field as part of the output record.

DOI: 10.1002/for

SECRET - INTERNAL PAGE: 0001

*** Sample DMS-100 "INVENTORY REPORT" Application ***

STOCK PART	LOCATION	QUANT	NLT	FUNCTION	ANALYSIS		EST	
					LIST	COST	1-VALUE	PRICE
907400	12-001-1	1750	1380	Dual 2-Inch WMD Gate	.19	110.0	143.18	75.46
907410	12-001-2	7300	2000	0 or 2-SpA WMD Gate (CC)	.19	111.72	222.32	95.40
907420	12-001-3	940	1000	Dual 2-Inch WMD Gate	.25	150.0	144.64	241.86
907430	12-001-4	773	800	Dual 2-Inch WMD Gate (CC)	.25	150.0	115.95	179.20
907440	12-001-5	612	600	WMD Inverter	.25	150.0	61.00	103.86
907450	12-002-1	296	300	WMD Inverter (CC)	.25	150.0	64.96	74.86
907460	12-002-2	29	100	WMD Inverter Buffer (CC-H)	.29	170.0	5.84	8.41
907470	12-002-3	940	900	WMD Buffer/Driver (CC-H)	.29	170.0	77.40	127.05
907480	12-002-4	296	300	WMD 2-Inch WMD Gate	.25	150.0	248.68	336.86
907490	12-002-5	296	300	WMD 2-Inch WMD Gate (CC)	.25	150.0	51.75	85.00
907500	12-003-1	1000	1100	1-2-Inch 2-Inch WMD Gate	.25	150.0	142.96	271.50
907510	12-003-2	917	900	1-2-Inch 2-Inch WMD Gate	.25	150.0	137.95	229.20
907520	12-003-3	1163	1200	1-2-Inch 2-Inch WMD Gate (CC)	.25	210.0	244.23	467.85
907530	12-003-4	40	60	Dual 4-Inch Schmitt Trigger	.25	210.0	60.28	147.86
907540	12-003-5	299	300	WMD Schmitt Trigger	.29	290.0	87.96	134.31
907550	12-004-1	247	300	WMD Inverter Buffer (CC-H)	.25	150.0	74.00	128.86
907570	12-004-2	1492	1600	WMD Buffer/Driver (CC-H)	.25	250.0	156.38	248.58
907580	12-004-3	222	230	Dual 4-Inch WMD Gate	.19	110.0	254.32	423.86
907590	12-004-4	480	750	Dual 4-Inch WMD Gate	.25	210.0	140.61	221.00
907600	12-004-5	696	500	4-Inch WMD Gate (CC)	.29	290.0	130.92	223.20
907620	12-005-1	296	300	Expandable Dual 4-Inch WMD Gate	.50	250.0	61.93	103.86
907630	12-005-2	611	600	Dual 4-Inch WMD Gate	.29	170.0	106.36	177.19
907640	12-005-3	700	800	Dual 2-Inch TTL/MS Interface	.29	170.0	137.28	228.81
907650	12-005-4	800	800	1-2-Inch 2-Inch WMD Gate	.25	200.0	80.45	137.75
907660	12-005-5	239	250	Dual 2-Inch WMD Buffer	.50	250.0	70.26	121.11
907670	12-005-6	100	100	Dual 2-Inch WMD Buffer	.25	150.0	167.15	246.25
907680	12-005-7	100	100	Dual 2-Inch WMD Buffer	.29	170.0	176.16	278.19
907690	12-005-8	100	100	Dual 2-Inch WMD Buffer	.25	150.0	155.75	240.25
907700	12-005-9	299	300	Dual 2-Inch WMD Buffer (CC)	.29	170.0	36.36	60.60
907710	12-006-1	290	300	Dual 2-Inch WMD Buffer (CC)	.50	250.0	61.28	113.67
907720	12-001-1	266	300	WMD Inverter	.19	110.0	50.82	76.84
907730	12-001-2	940	900	WMD-to-MS Decoder/Buffer Driver	.60	350.0	208.79	347.99
907740	12-001-3	295	300	WMD-to-MS Decoder	.50	270.0	51.57	85.75
907750	12-001-4	100	100	MS-to-WMD Decoder	.50	290.0	54.77	91.00
907760	12-001-5	192	200	MS-to-WMD Decoder	.50	210.0	114.44	194.86
907770	12-002-1	200	200	WMD-to-MS Decoder Driver	.60	310.0	124.41	207.49
907780	12-002-2	600	600	MS-to-WMD Decoder/Buffer Driver	.60	310.0	124.41	207.49
907790	12-002-3	600	600	MS-to-WMD Decoder/Buffer Driver	.60	310.0	124.41	207.49
907800	12-002-4	600	600	MS-to-WMD Decoder/Buffer Driver	.60	310.0	124.41	207.49
907810	12-002-5	600	600	MS-to-WMD Decoder/Buffer Driver	.60	310.0	124.41	207.49
907820	12-002-6	600	600	MS-to-WMD Decoder/Buffer Driver	.60	310.0	124.41	207.49
907830	12-002-7	600	600	MS-to-WMD Decoder/Buffer Driver	.60	310.0	124.41	207.49
907840	12-002-8	600	600	MS-to-WMD Decoder/Buffer Driver	.60	310.0	124.41	207.49
907850	12-002-9	600	600	MS-to-WMD Decoder/Buffer Driver	.60	310.0	124.41	207.49
907860	12-003-1	812	800	Exp. Dual 2-Inch 2-Inch WMD/2-Inch	.19	110.0	92.56	154.28
907870	12-003-2	812	800	Exp. Dual 2-Inch 2-Inch WMD/2-Inch	.19	110.0	92.56	154.28
907880	12-003-3	812	800	Exp. Dual 2-Inch 2-Inch WMD/2-Inch	.19	110.0	92.56	154.28
907890	12-003-4	812	800	Exp. Dual 2-Inch 2-Inch WMD/2-Inch	.19	110.0	92.56	154.28
907900	12-003-5	812	800	Exp. Dual 2-Inch 2-Inch WMD/2-Inch	.19	110.0	92.56	154.28
907910	12-003-6	812	800	Exp. Dual 2-Inch 2-Inch WMD/2-Inch	.19	110.0	92.56	154.28
907920	12-003-7	812	800	Exp. Dual 2-Inch 2-Inch WMD/2-Inch	.19	110.0	92.56	154.28
907930	12-003-8	812	800	Exp. Dual 2-Inch 2-Inch WMD/2-Inch	.19	110.0	92.56	154.28
907940	12-003-9	812	800	Exp. Dual 2-Inch 2-Inch WMD/2-Inch	.19	110.0	92.56	154.28

9044H	12-884-5	523	580 Dual In-Line Encoder	.19	.1148	59.62	99.37	37.75
9047H	12-885-1	721	788 Low In-Speed & Flip-Flop	.29	.1770	125.45	209.89	10.44
90472H	12-885-2	1892	1188 & Master/Slave Flip-Flop	.29	.1770	198.88	216.40	124.48
90473H	12-885-3	2822	2088 Dual & Master/Slave Flip-Flop	.35	.2181	424.62	787.76	283.88
90474H	12-885-4	1899	2088 Dual & Low Flip-Flop	.35	.2181	417.89	696.15	278.46

BIT BUCKET



Universal Data Systems

Information Systems Group

3000 Bradford Drive, Huntsville, AL 35895-1804

Phone (205) 837-8100 • Telex 810-756-2108

NEWS

Universal Data Systems drops price on direct connect line powered 300 baud modem

Universal Data Systems (Huntsville, AL), a subsidiary of Motorola, Inc. (Information Systems Group), now offers a lower price for the advantages of computer communications by telephone. The Universal Data Systems model UDS 103 O/A LP modem is available for \$145 (suggested US resale), a \$50 reduction from its original price, at participating Universal Data Systems dealers.

The UDS 103 O/A LP helps computer users communicate with each other as well as with dial-up computer information services, time-sharing systems, computer bulletin board services and more (additional charges and software may be required).

The UDS 103 O/A LP can thank advances in engineering technology (that greatly reduce its power requirements) for its sleek, low-profile design. Since it is powered entirely by the telephone line, there are no bulky, heat-producing power supplies or modules. The result is a trim, handsome unit that tests comfortably under a telephone.

For further information, contact Universal Data Systems, Inc., 3000 Bradford Drive, Huntsville, Alabama 35895; (205) 837-8100.

TECHNICAL SPECIFICATIONS: UDS 103 O/A LP

The UDS 103 O/A LP offers 0-300 bps phase coherent frequency shift keyed (FSK) communications under Bell 103/113 protocol. It is intended for use over two-wire direct-distance dial (DDD) public switched telephone networks.

Controls and Indicators: A talk/data switch controls whether a telephone or modem is connected to the telephone line and an originate/answer switch, which selects the appropriate data exchange operating mode. A front panel light indicates when the modem is off-hook.

Dimensions: 6-1/8 in. W by 9-1/2 in. D by 1-1/8 H.

Weight: 15 oz.

UDS modem functions are available in circuit-card form for OEM applications; UDS produces more integral modem cards for the data communications market than any other manufacturer -- more than a half million modems during the past ten years.

Modems manufactured by Universal Data Systems are warranted to perform as specified for an entire year, assuming proper application and no physical or electrical abuse. During that period, UDS will repair or replace any modem that suffers a malfunction.

to Computer Publishing Center
68 Micro Journal
3900 Cassandra Smith
PO Box 849
Hudson, TN 37543
Attn: Mr. R. W. Anderson

from Fulvio BUZZON
Via Polignote 1
ROMA 00124
ITALY

ROMA, AUG 23 1983

Dear Mr. Anderson

I first got wind of PL9 on your column and by your description I thought it was just what I was dreaming about since I don't remember when.

First thing the day after I sent the proper amount and .. Presto PL9 was in my hands. After transferring it to my computer I run a FLEX version a bit out of standard on 3 inch completely double density disk it run immediately with absolutely no problems. I must say, after using PL9 for some months now, that it is considerably better than my wildest dreams, better in fact than mindrush seems to think. They say it is primarily intended as an assembler substitute but I am using it now for ALL my programming which involves rather big programs. Its biggest strength is the capability of passing parameters any number, any size and the fact that you are not insulated from the details but you can insulate yourself, if you want, the way you want. I wrote them a letter drawing to their attention some minor bugs I thought I found and almost by return postage I got an envelope containing a reply apologizing for the delay(!) and a new version with all the problems resolved and a new really good manual and all ... for free!!

These are things which restore your faith in mankind.

My previous experience in this respect was a lecture on the facts of life in computing and that as I had already paid I was stuck with it. Anyway if I really needed a working compiler I could get the listings (only of the runtime package and you!! for the modest amount of 30000 so I could fix it myself).

I think the 6800 community has now "THE COMPILER" and if we were head and shoulders above the rest, we are walking now over their heads.

I wish to Mindrush all good things I can think of, and I hope they will be around forever doing good things to people.

I am developing some PL9 procedures to do REALLY FAST random files processing under FLEX at the expense of some memory. As soon as I am pretty sure there are no bugs I will put them in the public domain.

Thank you for your kind attention and keep up the good work.

Yours truly
F. Buzzon

F. Buzzon

LUCIDATA Ltd., PO BOX 124, CAMBRIDGE CB2 5EZ, ENGLAND



PRESS RELEASE

*** New Software for 6800 users ***

Lucidata announces the availability of the much omitted level 7 revision of its Pascal runtime system for the 6800 processor. The current revision, which has been available for the 6800 on both FLEX 9 and UniFLEX since the beginning of 1983, offers many additional important features over earlier revisions. All the popular UCSD type string handling procedures (CONCAT, POS, INSERT, DELETE, COPY, STR, CLEAR and additionally ADDCHAR and POSCHAR) have been included in the run-time system body and may be accessed from a Pascal program by simply declaring them as EXTERNALS. This method does not introduce any non standard syntax into the language which has been made even more compliant with the internationally recognised ISO standard. The current revision also introduces an auto restart option which may be selected by the user. In the event of an abnormal program termination the users program will be restarted and passed relevant information about the error condition as well as the number of times a restart has occurred. The user program is then free to decide whether it can safely recover.

The Pascal compiler (level N) available with this revision of the run-time system supports separately compiled modules using the DYERLAY procedure and external file names. FLEX names may be ASSIGNED to logical file names dynamically during program execution. Despite its increased size the compiler still runs on a minimal FLEX 2 system using the p-code PAGING technique pioneered by Lucidata in 1978 when it introduced the first Pascal system to run on small 6800 based systems.

This new revision of Lucidata Pascal (3.2.17), for the much neglected 6800 user, comes complete with manual and many demonstration programs on one 8" or two 5" standard FLEX format diskettes at \$150 inclusive postage worldwide.

VISA and MC accepted. FLEX and UniFLEX are trademarks of TSC.

UPGRADING A PERCOM BASED Microcomputer SYSTEM

Several months ago, I upgraded my old Percom 6809 based SS-50 buss system. Mainly, I wanted to use the more popular soft sector, formatted diskettes; especially the 80 track, double side/double density, 5 inch disk drives. The old Percom LFD-400 system uses hard sector formatted diskettes which have now become obsolete; especially, since Percom has

discontinued support for their SS-50 buss products. Since my memory and input/output boards and circuitry were still very useable, I decided to purchase, from AAA Chicago Computer Center, an Elecktra 6809 CPU board and a DCB-4A floppy disk controller board (made by Smoke Signal Broadcasting) along with my favorite disk operating system, Microware's OS-9. I also purchased two new Tandon TM-100-4 80 track, double side/ double density disk drives to go with the new computer setup. I have been more than satisfied. It WORKS very well and has performed admirably almost right from the start.

The Elecktra CPU board is a standard, no frills type of board. The board layout and design have been well thought out, and all parts are silkscreened for easy identification. The board features 6k bytes of EPROM area, 1k byte of RAM, a 6840 triple timer, and an optional 14411 IC baud rate generator. It does not have extended addressing nor a DAT circuit setup, and it doesn't have any input/output devices of its own (eg: MC6850 ACIA's). Thus, one must have another board which contains serial or parallel, input/output interfacing devices.

The Elecktra cpu board can be ordered with either a 1 mhz or a 2 mhz clock rate. In both cases, the board is assembled and tested using 2 mhz components. Thus, if you have a 1 mhz version, you can easily convert it to a 2 mhz version by replacing the crystal and two ceramic capacitors. All the other components are already 2 mhz rated devices.

The only detractor from the Elecktra CPU board is AAA's use of large, higher voltage 3/4 inch diameter, ceramic bypass capacitors. Although the capacitors are quite functional and do not hinder operation in any way, they do make an otherwise beautiful board look unappealing. A user must allocate two SS-50 buss slots to the cpu board, because the capacitors stick out so far from the board. Smaller, lower voltage capacitors would do much for its appearance and free up the extra buss slot on the motherboard. The Elecktra CPU board does work, and it works very reliably. As far as I'm concerned if something works, that's 99% of the game.

Smoke Signal Broadcasting's DCB-4A floppy disk controller board is a work of art. Just like the GIMIX 68 FDC board, it is crammed with integrated circuitry. After one has seen and used these FDC boards, one wonders what the other boards are missing. The board is designed to handle up to four floppy disk drives in any mix of 5" or 8" types. When I first received my FDC board, it didn't work. But Smoke Signal was absolutely wonderful. I express shipped my board to them on a Monday and by Thursday I had it back, all repaired and ready to go. Since then I have experienced no problems at all with it.

The OS-9 disk operating system I received was the new release version 1.2. It has a number of improvements over the older 1.1 version and can now handle pipes and I/O protocols better. I have noticed nothing at all wrong with the operating system. It works very well. I've transferred all of my old OS-9 software over to the new system and have had all of it work. Thus, any changes have not affected anything in the system that I've noticed. When I talked with Microware, they mentioned that changes in the system, especially in the disk input/output areas, may take a noticeably longer time to execute. I've yet to notice any perceptible time lag over the 1.1 version that I have. But then, if one modifies the disk device descriptor modules in the area of sector interleaving and step time rates, one could either really slow it down or

gain definite improvements.

The instructions that come with the Elecktra CPU board are more than adequate for getting it working. However, I advise that one should be quite familiar with setting up systems like this, as you could easily wind up with a mess on your hands. The instructions already assume that you know what memory addressing is, and configuring your system's memory map and the serial input/output devices to match is left up to you. It's possible that modifications to your particular system may require soldering and even some specific board modifications.

The DCB-4A FDC board, though, caused a problem in my particular system. I had to cut the jumper on W2-1. That particular jumper in my system apparently causes the 'Q' clock to wind up on the FIRQ line, with detrimental results. If you encounter any difficulties, you might give some careful attention to the W2 jumper area. I also have an additional jumper installed at W2-8 for my particular installation and setup. The DCB-4A board comes already setup at address \$F760 and this is right smack in the EPROM U24 memory area, so if your OS-9 version is like mine, make sure your CPU board's U24 EPROM is disabled. Also, having the FDC board located at \$F760 makes it very easy to switch over to Smoke Signal's DOS and Technical Systems Consultant's Flex™ DOS, if desired.

Although I purchased two new Tandon 80 track dual sided drives, I kept the old MPI 40 track Percom drives as drive /d0 and /d1, and installed the new Tandon drives as /d2 and /d3. This allows me to be able to read and write to both 40 track and 80 track, 5 inch diskettes without any difficulties. Since OS-9 allows one to change both execution and data directories to any other location one desires, I simply change over to the 80 track drives after booting up. The Tandon drives are like the MPI drives in that they use a taut-band head positioner mechanism for moving the read/write heads in and out to the different tracks. I prefer the taut-band method over other screw drive mechanisms, as one doesn't have to worry about the head positioning mechanism getting sticky or dirty or requiring cleaning and relubricating. However, taut-band type drives do tend to make more noise.

The other recycled boards in my system are several 16k and 32k byte RAM boards made by Digital Research Computers out of Garland, Texas. The basic input/output board in this system is a SP-1 Thomas Instrumentation Board which features 4 ACIA's, a 6840 timer, and 3 PIA devices. For those of you who are interested, I also have an oversized 8 volt 30 amp power supply to handle all those 2114 RAM chips. In order to get 56k of contiguous RAM I also have a simple 2114 8k ram board that I made and wirewrapped myself. The homemade ram board resides at the \$C000-\$DFFF memory range.

To be able to transfer the old Percom hard sector disk data over to the new system requires that one must have both the Percom LFD-400 disk controller and the new DCB-4A controller installed at the same time. Unfortunately, you must be able to get the Percom LFD-400 disk controller driver program module over to the new OS-9 system. I did this by downloading the module to my Ackerman Digital Systems computer system and then reloading the module back up to the new system. One could also possibly program an EPROM and install it in the spare rom socket or use the OS-9 debugger and manually reenter the program into the new system (To use the spare rom socket would require having the FDC board located in memory someplace other than \$F760). Another possibility, would be to program a

2708 EPROM with the driver module and install it in one of the Ram sockets available on the LFD-400 FDC board. Next you must set up a new special disk drive device descriptor module. I called mine /D4 even though it used a physical /d1 drive assignment (Make sure it's calling the Percom LFD-400 module and driver program module).

Performing the transfer required removing the 8k memory board, installing the Percom controller board (it must be configured for address \$C000), and disconnecting drive /d1 from the DCB-4A and running a separate drive cable between the Percom board and the /D1 drive. Next, copy all the programs over as needed. When done, all one has to do is reconfigure the system back to its normal setup condition.

Since I'm running both 40 and 80 track drives on the same system, I discovered a problem that could catch one unprepared. You must modify the RBF-device descriptor modules before using them for 80 track drives, especially the IT.DNS drive track density byte. My version of OS-9 was configured for regular 40 track 48tpi drives. 80 track drives are 96tpi devices. You must change this device descriptor byte in order to read the normal OS-9 80 track disks, otherwise you'll get read errors. The OS-9 system will allow you to format and use 80 track disks with a 48tpi IT.DNS byte but the rest of the world will be using disks with the IT.DNS byte set to 96tpi and you will wind up having a compatibility problem eventually. If you've gotten this far into changing your device descriptor modules you might also want to configure the rest of the module for 80 or 40 track double side and double density capability.

Having used the new upgraded system for several months, I am satisfied that it is a reliable and workable setup. I can recommend the Elecktre CPU board and Smoke Signal's DCB-4A FDC board as excellent products. They do work. I also can recommend Thomas Instrumentation's SP-1 board and Digital Research's RAM boards, as they have performed flawlessly for several years now. Usually memory and input/output boards never get mentioned in the magazines. They are more or less taken for granted.

Earl W. Bollinger
B & E Software Services
342 West McDowell Road
Phoenix, Arizona 85003
602-258-1376

THE MICRO WORKS

NEW PRODUCT RELEASE

MAGIGRAPH

The Micro Works is pleased to announce the release of a graphic development utility for the Radio Shack Color Computer. MagiGraph is for experienced Basic and assembly language programmers. The program simplifies the task of drawing highly detailed graphic characters, up to and including an entire high-resolution graphic screen. MagiGraph has a full set of logical operation and pixel manipulation functions that simplify the development of a character in all its different possible color and position combinations. The program offers nine animation buffers, which allow the various positions of an animated character to be tested and revised so that animation

blends together smoothly. Finally, the pixel codes of a graphic character can be listed in hexadecimal numbers to a printer, and the graphic screen can be saved on cassette tape or floppy disks to be used later by another program or revised with MagiGraph.

MagiGraph, written by Kevin Dooley, is available on cassette for \$34.95 (tax required); disk for \$39.95 (32k Extended Color BASIC required); and disk cartridge for \$44.95. Extensive documentation and plenty of sample programs are included in the package.

P.O. BOX 1110 DEL MAR, CA 92014 619-942-2400

Specialty Electronics, Inc.



NEW SOFTWARE ANNOUNCED BY SPECIALTY ELECTRONICS INC.

CHIPMAN

Specialty Electronics has just released Chipman game to run on 6801 terminal based computer systems running OS9: Level 1 or Level 11. Chipman has 2 different boards, three levels of play, and even needs the 100 10 high degree. Movement of your player is done from the keyboard using the '6' for up, '2' for down, '4' for right, and '8' for left.

Chipman requires 20k of memory & comes on either 5" or 3 1/2" floppy disks and costs for \$49.95. For more information call or write Specialty Electronics.

```

*****  Item Left 257  *****
10 *****
20 *****
30 *****
40 *****
50 *****
60 *****
70 *****
80 *****
90 *****
100 *****
110 *****
120 *****
130 *****
140 *****
150 *****
160 *****
170 *****
180 *****
190 *****
200 *****
210 *****
220 *****
230 *****
240 *****
250 *****
260 *****
270 *****
280 *****
290 *****
300 *****
310 *****
320 *****
330 *****
340 *****
350 *****
360 *****
370 *****
380 *****
390 *****
400 *****
410 *****
420 *****
430 *****
440 *****
450 *****
460 *****
470 *****
480 *****
490 *****
500 *****
510 *****
520 *****
530 *****
540 *****
550 *****
560 *****
570 *****
580 *****
590 *****
600 *****
610 *****
620 *****
630 *****
640 *****
650 *****
660 *****
670 *****
680 *****
690 *****
700 *****
710 *****
720 *****
730 *****
740 *****
750 *****
760 *****
770 *****
780 *****
790 *****
800 *****
810 *****
820 *****
830 *****
840 *****
850 *****
860 *****
870 *****
880 *****
890 *****
900 *****
910 *****
920 *****
930 *****
940 *****
950 *****
960 *****
970 *****
980 *****
990 *****
1000 *****

```

MSR 232-1632 P.O. BOX 341 © 1980, CHICAGO 75708

* OS9 is a trademark of Microware Inc.

NOT JUST ANOTHER DUMB ROUTINE

MTERM.CMD is a modem communications program which permits a FLEX(tm)-based 6809 system to transfer characters between the system console and a second serial port. "So what," you say, "that's just another dumb terminal routine." And so it would be if MTERM didn't have a couple of extra tricks up its sleeve.

First of all, MTERM makes itself memory-resident after the first call, so you don't have to have a disk access every time you switch between FLEX and

modem. Lines 44-58 of the program show how this step is accomplished by creation of a User Command Table (UCT).

When you type a command at the terminal FLEX checks the two bytes at \$CC12 for the address of a table of memory-resident commands (the UCT) before it looks on disk. A UCT entry consists of the name used to invoke the command, e.g., "MTERM"; a null character (\$00); and the two-byte absolute address of the routine's entry point. We can make a single-entry UCT simply by tacking this information onto the start of our program (lines 56-58). Lines 47 and 58 cause a pointer to this UCT to be loaded in the appropriate location in FLEX when the command is first called.

The technique described is useful when a single user-written command will be called repeatedly during an operating session, and it certainly saves time and reduces disk activity. There are a couple of points to watch, however. Don't confuse the UCT address (MTABLE) with the command routine's entry or transfer address (MTERM) - the latter is the one which must accompany the END directive (line 146). Also, bear in mind that the first call to MTERM will wipe out any previous UCT pointer. In most cases this won't matter, since FLEX will always reload a command from disk if it's not in the UCT.

The second interesting feature of MTERM is software-selectable baud rate. Both 300 and 1200-baud modems are common these days, and it's convenient to be able to change speeds without having to mess with DIP switches on an I/O card. This feature requires no hardware modification. It takes advantage of a frequently overlooked property of a number of serial communications chips, Motorola's 6850 ACIA among them.

We get so used to supplying a clock signal at 16 times the desired baud rate that we forget that these IC's will accept a x64 clock as well. In the case of the 6850, bits 1 and 0 of the control register determine the counter divide ratio:

BIT		DIVIDE RATIO
B1	B0	
0	1	16
1	0	64

If we supply a clock at 19.2 kHz and divide by 16, we get 1200 baud; divide by 64, and we get 300 baud. A convenient 19.2 kHz clock will be found on pin 24 of the SS-30 bus if the line hasn't been misappropriated for something else.

MTERM initializes the modem port on every call, so it's easy to select the configuration to give the desired speed. Lines 120-128 of the listing do that job by changing the operand of an LDA immediate instruction in the ACIA initialization sequence (line 77). Obviously this routine must reside in RAM!

The remainder of the program after initialization is in fact "just another" dumb terminal routine which transfers ASCII characters from one serial port to another (lines 85-107). The only keyboard character trapped is CTRL D, used to stop MTERM and return control to FLEX. If another character is more convenient, change the equate at line 37.

The general syntax used to invoke the MTERM command is as follows:

MTERM[, <speed>]

The square brackets as usual indicate an optional argument. To run at the speed previously set, simply type:

MTERM

To change baud rate, type:

MTERM 300

or MTERM 1200

Any other value (or non-hex character) will result in an error message. The program defaults to 300 baud initially.

Finally, lines 112 and 113 deserve comment, although they are not a logical part of MTERM. I have a couple of routines (MP.COMD and MPRINT.SYS) which redirect program output to the modem port, just as FLEX's P.COMD and PRINT.SYS do to the printer. The problem is that once MPRINT.SYS is in place I can't get anything out on my printer.

It seems that in order to speed things up P.COMD tries to avoid reloading the print drivers. If the byte at \$CCE4 isn't \$39, FLEX assumes that PRINT.SYS is already in memory and tries to use whatever code is there as a print driver. When the code turns out to be a modem port handler instead, the printer just doesn't print. Hence the replacement of \$39 at \$CCE4 - it's a bit ham-fisted, but it does ensure that PRINT.SYS will be there when I need it.

All in all, I think MTERM.COMD is a bit more than just another dumb terminal program. I hope you will, too.

- - -

David M. Rees-Thomas
VE MICROSYSTEMS
223 East 8th Street
N. Vancouver, B.C. V7L 1Y9

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34

```

```

NAM MTERM
* MTERM.CMD
*
* D.M. Rees-Thomas
* April 7, 1983
*
* Permits system to emulate a TTY terminal
* at a 6850 serial port. TIC/ASIC must be
* connected to a 1200 baud 16 clock signal.
*
* MTERM expects character echo from the
* on-line system and is transparent to all
* characters except CTRL D (004, E07),
* which returns user to FLEX (to TSC).
*
* Calling syntax:
*
* MTERM[, <speed>]
* where <speed> = 300 or 1200
*
* =====
*
* Modem and terminal port addresses
*
F7BC MODERN EQU 0F7BC Modem ACIA status
F7DB MODERN EQU MODERN+1 Modem data
F7FE TERMN EQU 0F7FE Terminal ACIA
F7FF TERMN EQU TERMN+1
*
* Change the value of the STOP character
* if CTRL D is required in communication
* with on-line system.

```

```

35          *
36          0004 STOP EBU 004 CTRL 0 to exit
37          *
38          * FLEX equates
39          *
40          C003 WARMS EBU 0C003
41          C01E PSTRMG EBU 0C01E
42          C042 GETMEX EBU 0C042
43          *
44          * Set up pointer to User Command Table.
45          *
46          CC12          ORG 0CC12
47          CC12 F000          F00 USRTBL
48          *
49          * Set up user command table at start
50          * of program to make MTERM memory-
51          * resident after initial call.
52          *
53          F000          ORG 0F000
54          F000 00 04 45 52 USRTBL FCC 'MTERM'
55          F004 00          F00 000
56          F006 F000          F00 MTERM
57          *
58          * Program resides in an out-of-the-way
59          * corner of spare memory.
60          *
61          F000 20 01 MTERM 000 START
62          F004 02 VN FCB 002
63          F008 00 C042 START JSR GETMEX Speed entered?
64          F00E 25 67 BCS NSERR Not in valid hex
65          F010 26 50 BNE NSPEED OK, go set speed
66          *
67          * The modem port ACIA is initialized
68          * originally at 300 baud (divide by 64)
69          * with 052 in the control register.
70          * To run at 1200 baud, this byte is
71          * changed to 051 (divide by 16).
72          *
73          F012 06 03 MINIT LDA 0003 Initialize modem
74          F014 07 F70C STA MODEMS ACIA
75          F017 06 52 MCF LDA 0052 Set 300 (div=64)
76          F019 07 F70C STA MODEMS
77          *
78          F01C 30 00 0060 LEAI MODMSG,PCR In() operator
79          F020 00 C01E JSR PSTRMG we're on-line
80          F023 30 00 0073 LEAI CTMSG,PCR and how to quit
81          F027 00 C01E JSR PSTRMG
82          *
83          F02A 06 F70C CHKRDH LDA MODEMS Char. received
84          F02D 44 LSRA from modem?
85          F02E 24 0F BCC CHKTRM
86          F030 06 F70D LDA MODEMD Yes, send to term.
87          F033 04 7F ANDA 007F Strip parity first
88          F035 F6 F7FE LDB TERMS
89          F03B 34 LSRL
90          F039 54 LSRL
91          F03A 24 F9 BCC #5 TONE?
92          F03C 07 F7FF STA TERM0 Send char.
93          F03F 06 F7FE CHKTRM LDA TERMS Char. received
94          F042 44 LSRA from keyboard?
95          F043 24 E5 BCC CHKRDH No, try modem
96          F045 06 F7FF LDA TERM0 Yes, get character
97          F048 04 7F ANDA 007F Strip parity
98          F04A 01 04 CMAA 0104 Do we quit?
99          F04C 27 0C BEQ EXIT
100         F04E F6 F7FE LDB TERMS No, send to modem
101         F051 34 LSRL
102         F052 54 LSRL
103         F053 24 F9 BCC #5 TONE?
104         F055 07 F70D STA MODEMD Yes, send char.
105         F05B 20 00 BNA CHKRDH
106         *
107         * Next two lines are not necessarily
108         * required, but retain label EXIT.
109         *
110         F05A 06 39 EXIT LDA 0039 Replace RTS to let
111         F05C 07 CCE4 STA 0CCE4 PRINT.SYS reload
112         F05F 7E C003 JMP WARMS Return to FLEX
113         *

```

```

114          * Check rate entered and set speed.
115          * 300 and 1200 baud are the only
116          * rates permitted.
117          *
118          F062 0C 1200 NSPEED CMP1 001200 1200 baud?
119          F065 26 04 BNE NSLOW Evidently not
120          F067 04 51 LDA 0051 1200: divide by 16
121          F069 20 07 BNA SETSPB
122          F06B 0C 0300 NSLOW CMP1 00300 300 baud?
123          F06E 26 07 BNE NSERR Nope, invalid
124          F070 06 52 LDA 0052 300: divide by 64
125          F072 A7 0C A3 SETSPB STA MCF+1,PCR Config. ACIA
126          F075 20 90 BNA MINIT Run as terminal
127          *
128          * Error if speed not 300 or 1200.
129          *
130          F077 30 00 0042 NSERR LEAI MSENSB,PCR Invalid speed
131          F07B 00 C01E JSR PSTRMG
132          F07E 20 0A BNA EXIT
133          *
134          F080 0A 00 MODMSG FCB 00A,000
135          F082 20 20 20 20 FCB '*** MTERM ***'
136          F086 20 20 20 20 FCB 00A,000,004
137          F08A 50 34 79 70 CTMSG FCC 'Type CTRL D to'
138          F09E 45 20 43 54 FCB 00A,00A,00D,004
139          F0A2 52 4C 20 44 FCB 'Speed 300 or 1200'
140          F0A6 20 74 6F FCB 'baud only **'
141          F0AA 20 72 65 74 FCB 007,004
142          F0AB 75 72 6E 20 FCB 007,004
143          F0B1 74 6F 20 46 FCB 007,004
144          F0B5 4C 45 58 50 FCB 007,004
145          F0B9 0A 0A 00 04 FCB 007,004
146          F0BB 2A 2A 20 53 FCB 007,004
147          F0C1 70 65 65 64 FCB 007,004
148          F0C5 20 33 30 30 FCB 007,004
149          F0C9 20 6F 72 20 FCB 007,004
150          F0CB 31 52 30 30 FCB 007,004
151          F0D1 20 62 61 75 FCB 007,004
152          F0D5 64 20 6F 6E FCB 007,004
153          F0D9 6C 79 20 2A FCB 007,004
154          F0DD 2A FCB 007,004
155          F0DE 07 04 FCB 007,004
156          *
157          END MTERM

```

0 ERROR(S) DETECTED

SYMBOL TABLE:

```

CHKRDH F02A CHKTRM F03F CTMSG F09A EXIT F05A GETMEX C042
MCF F017 MODEMD F70D MODEMS F70C MODMSG F000 MTERM F000
NSERR F077 NSLOW F060 NSPEED F062 MTERM F000 MINIT F012
PSTRMG C01E SETSPB F072 START F00B STOP 0004 TERM0 F7FE
TERMS F7FE USRTBL F000 VN F00A WARMS C003

```

"HOLD" REVISED

2714 Thosom
Flint, MI 48504
18 AUG 1983

Mr. Don Williams
68 Micro Journal
Box 049
Mixon, TN 37343

Don:

While reviewing some back issues recently, I discovered Rick Scappatura's "HOLD" program (May 82 p. 25). I thought that this would be a useful utility so I typed it in.

Though the idea was good, I found the implementation less than perfect for my needs. As written, the program tries to get the "hold" program off of the work drive instead

of the system drive where most of my programs live. This does not present a problem if the automatic search option is on (i.e. ...ASM.W.A), but say if the working drive is set to a specific unit.

It only took a couple of added lines to make HOLD work the way I thought it ought. The attached listing shows my version of the program. The problem occurs because HOLD uses FLEX's "GETFIL" subroutine to parse the name of the program to be run. In the absence of a drive specifier, GETFIL takes the working drive as a default. Since I wanted it to use the system drive instead, I merely fool it by temporarily setting WRKDRV to be the same as SYSDRV. This assignment is made immediately before the call to GETFIL, and is restored upon return from that routine.

I find that the program is such a useful now. I hope that your readers will too.

Sincerely,

Donald H. Korte

Donald H. Korte

```

!
! program HOLD is used to suspend
! execution of a FLEX command until
! the user signifies that he is
! ready. To run the FLEX command
! "command", the user types:
!
!   ++HOLD,command
!
! the program called "command" is
! loaded off the system disk, then
! a prompt is printed. When the
! user is ready, he signifies
! such by typing "Y". The command
! is then run.
!
! This program taken from an original
! by Rick Scappatura printed in
! 68 Micro Journal, MAY 82, p.25 .
!
! modified by DN Korte 8/8/83
! to take "command" from the system
! disk.
!
! FLEX equates

```

```

C840 SYSFCB equ $C840
CC10 LDRFF equ $CC10
CC10 XFERF6 equ $CC10
CC1E XFERAD equ $CC1E
CD03 WARMS equ $CD03
CD15 BETCHR equ $CD15
CD18 PUTCHR equ $CD18
CD1E PSTRNG equ $CD1E
CD20 GETFIL equ $CD20
CD30 LOAD equ $CD30
CD33 SETEXT equ $CD33
CD3F RPTERR equ $CD3F
CD24 PCRLF equ $CD24
D403 FMSCLS equ $D403
D406 FMS equ $D406
CC0C WRKDRV equ $CC0C
CC0B SYSDRV equ $CC0B

```

```

F600 org $F600
F600 20 01 hold bra h1
F602 01 vn fcb 1
F603 06 CC0C h1 lda WRKDRV
F606 34 02 pshs a

```

version number
get crnt wkg drv &
save it &&&&

```

F608 06 CC0B lda SYSDRV
F608 07 CC0C sta WRKDRV
F60E 0E C840 ldx $SYSFCB
F611 0D CD20 jsr GETFIL
F614 35 02 puls a
F616 07 CC0C sta WRKDRV
F619 25 52 bcs err2
F61B 06 02 lda #2
F61D 0D CD33 jsr SETEXT
F620 25 3C bcs err1
F622 CC 0000 ldd #0000
F625 FD CC1B std LDRFF
F62B 06 01 lda #1
F62A A7 04 sta 0,x
F62C 0D D406 jsr FMS
F62F 26 2D bne err1
F631 06 FF lda #FF
F633 A7 0B 30 sta 59,x
F636 0D CD30 jsr LOAD
F639 7D CC1D tst XFERF6
F63C 27 27 beq err3

```

current system dri
make wkg drv same
now restore &&&&
working drive &&&&
set for CMD
loader offset
open for read
do it...
set space flag
for binary
get file
check transfer add
no exist

```

! continue ???
F63E BE F6B4 what ldx #msg3
F641 0D CD1E jsr PSTRNG
F644 0D CD15 jsr BETCHR
F647 04 0F anda #DF
F649 01 59 capa #Y
F64B 27 0A beq go
F64D 01 4E capa #N
F64F 26 ED bne what
F651 0D CD24 jsr PCRLF
F654 7E CD03 jmp WARMS
F657 0D CD24 go jsr PCRLF
F65A 6E 9F CC1E jmp XFERAD

```

y or n
get y or n
make upper
yes??
yep
no??
nope
if N then quit
go to it!

```

! errors here
F65E A6 01 err1 lda 1,x
F660 0D CD3F jsr RPTERR
F663 20 0E bra quit
F665 BE F691 err3 ldx #msg2
F668 0D CD1E jsr PSTRNG
F66B 20 06 bra quit
F66D BE F679 err2 ldx #msg1
F670 0D CD1E jsr PSTRNG
F673 0D D403 quit jsr FMSCLS
F676 7E CD03 jmp WARMS

```

```

! messages
F679 2D 2D 20 69 msg1 fcc '-- illegal command name',4
F691 2D 2D 20 6E msg2 fcc '-- no transfer address for
F6B4 2D 2D 20 63 msg3 fcc '-- continue Y/N ? ',4
!
end hold

```

SYMBOL TABLE:

FMS	D406	FMSCLS	D403	BETCHR	CD15	GETFIL	CD20	LDRFF	C
LOAD	CD30	PCRLF	CD24	PSTRNG	CD1E	PUTCHR	CD1B	RPTERR	C
SETEXT	CD33	SYSDRV	CC0B	SYSFCB	C840	WARMS	CD03	WRKDRV	C
XFERAD	CC1E	XFERF6	CC1D	err1	F65E	err2	F66D	err3	F
go	F657	h1	F603	hold	F600	msg1	F679	msg2	F
msg3	F6B4	quit	F673	vn	F602	what	F63E		

Dear Editor,

We at Quest Computers Inc. would like to take this opportunity to introduce our new software package, VIDIX - The Video Rental Management Program you mentioned in the July issue of "68 Micro Journal". This package, which runs on Flex and Uniflex, which has taken almost two years of hard work and testing is now complete. We have installed systems in every major video store in Milwaukee and they have been up for several months.

The VIDIX system organizes and monitors the store's activities and is available for a single store or a series of stores. Among its many features are:

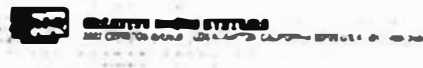
- Opening and closing reports
- Commissions for new member sign
- Non-member rental protection
- Pre-recorded tape stock checks
- Tape traffic analysis by time of day and category
- Bar code technology
- Locks out delinquent club members

and much much more.

Demo packages are available for \$35.00 and soon there will be a demo available on video tape!

Readers and dealers can call or write for pricing and literature. The response we have been getting for this product has been phenomenal, and there is really no competition!

Quest Computers Inc.
4680 W. Bradley Rd.
Brown Deer, WI. 53223



PROGRAMMING HEAD ACCOMMODATES 64K AND 128K EPROMS

THE 9618 EPROM PROGRAMMING HEAD WITH SUPPORT FIRMWARE IS A UTILITY MODULE FOR USE WITH 6800 SYSTEMS. IT PROVIDES A SIMPLE MEANS FOR PROGRAMMING SINGLE-VOLTAGE EPROMS FROM DATA RESIDENT IN THE SYSTEM MEMORY. THE 9618 IS DESIGNED TO DERIVE ALL SIGNALS AND POWER DIRECTLY FROM THE PARALLEL I/O CONNECTOR, P3, OF ANY CMS SINGLE BOARD MICROCOMPUTER. IT CAN ALSO BE INTERFACED TO ANY SYSTEM WHICH PROVIDES TWO UNBUFFERED MC6821 PARALLEL INTERFACE ADAPTERS (PIA'S), AND A SOURCE OF +5 VDC. THE 9618 IS INTENDED FOR TABLETOP USE AND IS CONNECTED TO THE SYSTEM I/O PORT BY A 50 CONDUCTOR FLAT CABLE.

THE 9618 IS SUPPORTED BY FIRMWARE THAT PROVIDES CONVENIENT OPERATOR CONTROL OVER THE PROGRAMMING PROCESS. SINGLE VOLTAGE 28PIN 2764 AND 27128 EPROMS CAN BE READ, TESTED FROM ERASED CONDITION, PROGRAMMED AND VERIFIED. DATA IN MEMORY CAN BE INSPECTED AND CHANGED PRIOR TO PROGRAMMING.

THE SINGLE QUANTITY PRICE FOR THE 9618 EPROM PROGRAMMING HEAD WITH SUPPORT FIRMWARE IS \$295.00. QUANTITY DISCOUNTS ARE AVAILABLE. DELIVERY IS FROM FACTORY STOCK.

ADDITIONAL INFORMATION IS AVAILABLE FROM CREATIVE MICRO SYSTEMS, 3822 CERRITOS AVENUE, LOS ALAMITOS, CA 90720. TELEPHONE (213) 493-2484.

A SMALL C

Normally we do not reproduce an instruction book as a review. Fact is, we attempt to keep this from happening. Too many times we all see 'reviews' in other magazines that are nothing but excerpts of some products documentation. This sort of 'review' cannot get to the real world conditions of the product. So, we normally stay away from that type 'reviewing'!

However, in this case we are reproducing an instruction manual as a means of letting those 6800 users (oft forgotten) who want to try it, at a very reasonable price, an opportunity to see what the author has done. Please note the advertised price in the next few issues of 68 MICRO JOURNAL.

The sTs Small-C is an improved version of a C subset originally published in Dr. Dobbs Journal, as presented

by Ron Cain, titled A Small C Compiler. Originally done for the 8080, Serge Stepanoff of sTs has rewritten the code to take advantage of the MC6800. The sTs version runs under TSC FLEX™ and is to my knowledge the first to do so.

Versions have been written for practically every other CPU including the MC6809. And for those who are still using the 6800, this will enable them to begin to use the C language, which I feel will be the primary programming language in the future.

I am told that if enough (don't know what enough is) of you desire, sTs will do a version 2. I am sure that this will depend on the response they receive for the present version. As I feel that there is a need for support yet for the 6800, I trust you who still use the 6800 will appreciate additional software availability for your systems.

sTs ENTERPRISES

6069 ARIZONA WAY, LIVERMORE, CA 94550
(415) 466-6750

Version 2 for FLEX: FLEX is a trademark of Technical Systems Consultants
Serge Stepanoff
6069 Arizona Way
Livermore, CA 94550

INTRODUCTION

Small-C is a subset of the C language — a subset of the language C developed at Bell Labs and later adopted by the standards committee. It is the C Programming Language, by Brian Kernighan and Dennis Ritchie, published by Prentice-Hall, Inc., 1978.

The original Small-C compiler was written by Ron Cain for the 8080 microcomputer on a North Star system. He was glad and astonished enough to place it in the public domain via articles in "Dr. Dobbs Journal". Since then, this compiler has been adapted to several operating systems and microcomputers, including the 6800. To this author's knowledge, this is the first adaptation for the 6800 micro under TSC's FLEX operating system. The project finished approximately one year's worth of calendar time, mainly because the work on it was carried out in a sporadic and periodic basis.

Initial conversion attempts involved conversion directly to assembly language, but this approach was quickly dropped due to lack of any 6800 arithmetic in the 6800 instruction set and general paucity of stack operations. Assembly code was tedious to do, too, too. As for the 6800, it is about, however, in its tradition to PASCAL's 68000 and 68010, threaded code, a solution was found in translating to a pseudocode and then interpreting this code on a virtual machine. The translator in execution time is much less than for a straight interpreter like BASIC — the pseudocode interpreter and the run-time library occur less than 30 bytes, and as an interesting side note, to execute this pseudocode on a different operating system or a different micro-computer requires only the rewriting of the interpreter and the run-time library for the target machine.

As a historical note, the initial conversion was not done on a UNIX system, but rather on a PDP-11 running RS-11 operating system and the DECUS version of a C compiler (another Public-domain User Group product) with the Small-C code supplied by DECUS (Digital Equipment Corporation Users' Association). The resulting development version of the Small-C compiler is being carried out on a SUPTR 6800 system with dual 8 inch single density floppy drives, and 32K of RAM memory.

OPERATIONS

The Small-C compiler is invoked by typing CC on the keyboard. No command line arguments are necessary. A series of questions are then asked. After each question, Small-C prints in parentheses what the possible responses are. The conventional response to the one that Small-C will default to if you just press RETURN. The first question is:

Do you want the C-text to appear (Y/N) ?

This gives you the option of introducing the source code into the output file. Responses Y or N or y or n will also serve. If Y is given, an asterisk will be placed at the start of each input line to force a comment to the 6800 assembler and the input lines will be interleaved with the compiler generated pseudocode output. If the answer is N (or just RETURN, since N is capitalized, and is the default), only the generated pseudocode will be output. NOTE: the TSC assembler accepts any length labels, both lower and upper case, but only the first 8 characters are used and saved in the symbol table. Therefore, if you have either functions or labels of the type MODALES and MODALES, the assembler will generate a multiply defined label error. Be sure that the first 8 characters are unique. C-att is so.....

Are you compiling the whole program at once (Y/N) ?

This is a convenience (or annoyance) question. If you type Y (or RETURN), you will be able to avoid three more tedious questions, which are only needed if you, C program will be fed into Small-C in several separate compilations. (Just say so to it.)

If you answer N to the all-at-once question, you'll also be asked:

Do you want the globals to be defined (Y/N) ?

This question is primarily a developmental aid between machines. If the answer is Y, all static symbols will allocate storage within the module being compiled. This is the normal method. If N (or RETURN), no storage will be allocated, but symbol references will still be made in the normal way. Essentially, this question allows the user to specify all or none of the static symbols 'external'.

Is the output file the first one the assembler will see (Y/N) ?

If it is, Small-C will put out a short prototype that places the command line parameters (if any), sets up the stack, and calls the C program as a subroutine, so that a return to FLEX may be made properly when the routine is finished.

Starting number for labels (0) ?

This lets you supply the first label number generated by the compiler for its internal labels (which will eventually be "global", where "global" is a decimal number). This option allows modules to be compiled separately and later amended on the source level without generating multi-defined labels. If you just press RETURN, labels will start at 00000.

Now a question that requires some consideration:

Should I pause after an error (Y/N) ?

If you are doing a long compilation, the part that you go off and see the loan during, you can the risk of having an error message scroll off the screen

'68' Micro Journal

All C Compilers are Not Created Equal!



**Compare Introl-C/6809
with any other C being offered
for the MC6809.**

**The differences may really
surprise you!**

Introl-C is a powerful software development tool designed for the professional. It supports the full language, is truly easy to use, and produces remarkably efficient object code for the 6809. In fact, code generated by Introl-C typically is only **half the size and twice as fast** as code produced by other C compilers on the market! As we said, all C compilers are not created equal. Introl-C/6809 delivers what others only promise.

Resident Introl-C compilers for:
OS9*, FLEX, and UnifLEX**, \$375.**

Cross compilers for:
PDP-11* hosts (Unix****), \$1500.**

Trademarks:

*Microware Systems, **Technical Systems Consultants,
Digital Equip. Corp., *Bell Labs.

INTROL
CORPORATION

847 W. Virginia St.
Milwaukee, WI 53204
(414) 278-2837

management utilities. This will permit EXORmcs systems with 8" floppy disk drives to boot with CP/M-68K, and to execute programs designed to operate under the CP/M-68K environment.

The VERSAdos disk format will be particularly useful to OEM system designers who are creating products that use the Motorola MC68000 processor with the CP/M-68K Operating System. The VERSAdos disk format will be available on 8" floppy, 16" SMD, and 8" Lark disk media. It will allow an EXORmcs Development System owner to load the CP/M-68K object module onto his development system. He can then customize, download, and debug his system with standard Motorola development system languages, assemblers, linkers, emulators, and symbolic debuggers.

The CP/M-68K Operating System will be distributed on Motorola EXORmcs Development System Products under an agreement with Digital Research Inc., of Pacific Grove, CA.

Motorola will discuss and demonstrate both UNIDOS and CP/M-68K™ in Motorola's display booth at the WESCON trade show in mid-September.

CP/M and CP/M-68K are trademarks of Digital Research, Inc.

EXORmcs, UNIDOS, and VERSAdos are trademarks of Motorola Inc.

MOTOROLA SELECTS LANGUAGE RESOURCES'

68000 MICROPROCESSOR PRODUCT SUPPORT LINE

BOULDER, Colorado — Motorola has contracted to become a non-exclusive stocking distributor for Language Resources' cross-software package for the 68000 16-bit microprocessor, both companies announced.

The Language Resources 68000 cross support system (XSB-68), first introduced in early 1982, will also continue to be available directly from the Boulder-based firm, and from Signotice, Inc., a 68000 microprocessor award source and previously-appointed distributor, according to Geoffrey Archibald, Language Resources' President.

Language Resources is a leading innovator in advanced systems software products, and is widely known for its work in solving language barrier problems between main frame computers and microprocessors.

"After careful consideration of our customers' needs and evaluation of 68000 product support lines, there was no question in our minds that Language Resources' XSB-68 package met the requirements and was the way to go," was how Michael Hagerty, Motorola Microsystems, Tempe, Arizona, explained the decision to stock Language Resources' cross-software package.

On his part, Archibald said, "We are pleased that our 68000 product line continues to gain the recognition we always felt it deserved, and pleased it has performed as expected in thousands of industrial quality programs. User experience has shown that all of its elements work well together and it is easy to use."

In its 68000 product support line, Language Resources provides the tools that enable engineers to use main frame computers such as DEC's VAX and IBM's 370 and compatible systems, and do away with the limitations and constraints of typical microprocessor development systems. Because the use of a main frame computer offers a broad choice of software tools, greater speed and nearly unlimited storage capacity, the end result is a development system that simplifies microprocessor software implementation and maintenance.

The 68000 product line consists of:

- A high-level Pascal compiler;
- A powerful macro assembler;

CLASSIFIED ADVERTISING

FOR SA E: Shugart 5MB SA-604 Winchester drives for \$275. Very low hours. Kirk Thomas, 7608 Halsey #306, Lenexa, KA 66216 (913)268-9445

SWTP 6800 / 6809 running FLEX. complete system, lots of software, minus drives \$1500. Will sell pieces if necessary. Bill (716) 232-3448.

Gimix 6809 system, dual DS DD 5 1/4" disks, 4 serial ports, parallel port, OS-9, FLEX. Lots of software, \$3950. Additional 64K RAM board \$250. Dual 8" disks \$750. Other SS-50 stuff. Phil Hughes (206) 367-8649

Wanted: bare SWTPC MP-A2 and MP-N boards. Bill Meral (303) 597-4575

For Sale: SWTPC 6809 System. UniFlex OS. Includes: 128K Ram; 20 Megabyte Hard Disk; two 8212 terminals, Dual 8" DSD drives; Miscellaneous Software, Furniture (Desk, Stand). Less than 1000 hours on time. 1 1/2 years old. Excellent Condition. Make me an offer Call Marilyn (512) 250-9844, Business Data Systems, 8300 Pilgrims Pl., Austin, TX 78759

CoCo Software (original w/doc): Dynasoft Pascal (FLEX) \$45; ccFORTH \$60; 30 Drawing Board \$15; \$10 each: Master Control, MDISK, Magic Box, Fixer, Videotex Hardware: P180C \$25, CPRINT \$25, Video Plus \$15. Lane Lester, 413 Woodland Circle, Lynchburg, VA 24502

Southwest Technical: 8212 terminal; 6809 C.P.U. with 64K; much flex software; 2 Qume 8" D.D. Disk Drives - \$2,000.00 OBO Call Bill - (805) 963-1453

CONTINUED ON PAGE 56

COMPIER EVALUATION SERVICES By: Ron Anderson

The S.E. MEDIA Division of Computer
Publishing Inc.
Is offering the following **SUBSCRIBER
SERVICE:**

COMPILER COMPARISON AND EVALUATION REPORT

Due to the constant and rapid updating and enhancement of numerous compilers, and the different utility, appeal, speed, level of communication, memory usage, etc., of different compilers, the following services are now being offered with periodic updates.

This service, with updates, will allow you who are wary or confused by the various claims of compiler vendors, an opportunity to review comparisons, comments, benchmarks, etc., concerning the many different compilers on the market, for the 6809 microcomputer. Thus the savings could far offset the small cost of this service.

Many have purchased compilers and then discovered that the particular compiler purchased either is not the most efficient for their purposes or does not contain features necessary for their application. Thus the added expense of purchasing additional compiler(s) or not being able to fully utilize the advantages of high level language compilers becomes too expensive.

The following COMPILERS are reviewed initially, more will be reviewed, compared and benchmarked as they become available to the author:

PAS AL. "C" GSPL WHIMISCAL PL/9

Initial Subscription - \$39.95
(Includes 1 year updates)
Updates for 1 year - \$14.50

S.E. MEDIA - CPI
5900 Cassandra Smith, POB 794
Hixson, TN 37343
615 842-4601

DYNAMITE+™

"THE CODE BUSTER"

disassembles any 6809 or 6800
machine code program into beautiful source

- Learn to program like the experts!
- Adapt existing programs to your needs!
- Convert your 6800 programs to 6809!
- Automatic LABEL generation.
- Allows specifying FCB's, FCC's, FDB's, etc.
- Constants input from DISK or CONSOLE.
- Automatically uses system variable NAMES.
- Output to console, printer, or disk file.
- Available for all popular 6809 operating systems.

FLEX™ \$100 per copy; specify 5" or 8" diskette.

OS-9™ \$150 per copy; specify 5" or 8" diskette.

UNIFLEX™ \$300 per copy; 8" diskette only.

For a free sample disassembly that'll convince you DYNAMITE + is the world's best disassembler, send us your name, address, and the name of your operating system.

Order your DYNAMITE+ today!

See your local DYNAMITE + dealer, or order directly from CSC at the address below. We accept telephone orders from 10 am to 6 pm, Monday through Friday. Call us at 314-576-5020. Your VISA or MasterCard is welcome. Orders outside North America add \$5 per copy. Please specify diskette size for FLEX or OS-9 versions.

Foreign Dealers:

Australia & Southeast Asia: order from Paris Radio Electronics, 161 Bunnerong Road (PO Box 380) Kingsford, 2032 NSW Australia. Telephone: 02-344-9111.

United Kingdom: order from Compusense, Ltd., PO Box 169, London N13 4HT. Telephone: 01-882-0681.

Scandinavia: order from Swedish Electronics hk AB, Murargatan 23-25, Uppsala S-754 37 Sweden. Telephone: 18-25-30-00.

Computer Systems Center
13461 Olive Blvd.
Chesterfield, MO 63017
(314) 576-5020



UNIFLEX software prices include maintenance for the first year.

DYNAMITE + is a trademark of Computer Systems Center.

FLEX and UNIFLEX are trademarks of TSC.
OS-9 is a trademark of Microware and Motorola.

Dealer inquiries welcome.

SOUTH EAST MEDIA
5900 Cassandra Smith Rd., Hixson, TN 37343
(615) 842-4601

Software

For Ordering Call TOLL FREE 1-800-338-6800 **FLEX™ OS-9™ Color Computer**

--- DISASSEMBLERS ---

Super Sleuth -- Powerful, INTERACTIVE, Disassembler; provides Hex/ASCII Screen Dump of Memory Blocks or Disk Files for easy Examine/Change, Area Definitions; X-REF Generator of Source Code; change Address Labels to Names; etc. Disassembles 6800/1/2/3/5/8/9 and 6502 Code.

Color FLEX Obj. Only \$50.00, w/source \$99.00; FLEX \$99.00; UniFLEX \$100.00; OS-9 \$101.00

DYNAMITE + -- An "easy to use" 6809 Disassembler for use w/ Disk Files, (will also disassemble 6800 Code). Develop a "Control File" of Area Definitions during successive disassemblies; X-Ref Source Files; replace Hex locations with Label Names; etc. Label Files provided for Mini-FLEX, FLEX2, FLEX9, Color Computer FLEX.
FLEX and Color FLEX \$100.00 OS-9 \$150.00 UniFLEX \$300.00

--- ASSEMBLERS ---

TSC Macro Assembler -- the FLEX STANDARD Assembler FLEX and Color FLEX \$50.00

TSC Reloc. Asmb./Link. Load. -- Need for many of the C and Pascal Comp. FLEX & Color FLEX \$150.00

RRMAC -- Relocating, Recursive-Macro Assembler and Linking Loader for the 6809. Use either standard Motorola Format or Special Ed Smith Format. Supports Recursive Macros, Conditional Assembly, etc. Opt. X-Ref Listing; Includes a Small Line-oriented Editor as part of the Assembler. Greatly improved operating manual. FLEX and Color FLEX 6809 RRMAC w/Link and Editor \$150.00 w/Source, add \$50.00

OmegaSoft Relocatable Assembler and Linking Loader -- 2-pass Reloc. Asmb; 2-pass Link. Load. Supports 6 Char. Labels, System Calls (SWI[x] FCB x), Expressions with Arith., Logic, and Shifts, etc.
#RALLI FLEX and Color FLEX \$125.00 (one year Maint. \$50.00)

MACE -- (by Graham Trott) from WINDRUSH MICRO SYSTEMS. A combined Editor/Assembler designed to allow the Programmer to Enter, Edit, and Assemble Programs with a minimum of effort, w/o leaving the Program. ~~Y~~MACE is a ~~Cross~~ Assembler for the MC6800/1/3 and Hitachi HD6301 (CMOS 6801) with the same functions and features as MACE.
FLEX and Color FLEX - \$98.00

---- A UniFLEX "basic" Compiler ----

DUB -- Re-Create a Source Listing from UniFLEX Compiled BASIC Programs. Easy to Use; works w/ ALL Versions of UniFLEX basic; Output to Disk or Term. Time TESTED and PROVEN; SOLID! UniFLEX \$219.95

--- COMPILERS ---

PL/9 -- (by Graham Trott) from WINDRUSH MICRO SYSTEMS. A "Structured" Assembly Language Editor/Compiler/Debugger, all in ONE PACKAGE; provides a totally INTERACTIVE Program Development Cycle. The Compiler supports large Symbol Names, Variable Types, Pointers, Control Structures, Stack, A-, B-, and D-Register manipulation, etc. The Source oriented Trace/Debugger provides Single Stepping, Breakpointing, etc. An excellent Software Development Tool for utilizing the power of the 6809.
FLEX and Color FLEX - \$198.00

C -- (By James McCosh) from WINDRUSH MICRO SYSTEMS. SUPER C Compiler for the FLEX Operating System. Needs the TSC Relocating Assembler/Linking Loader for those "full blown" System Packages.
FLEX and Color FLEX - 295.00

Introl 6809 "C" Compiler; generates very efficient object code. Output "benchmarks" close to 10MHz 68000 in 8 Bit Operations; 1.5 times faster than a 4 MHz Z80 when using a 2MHz 6809 System (Re. p 43, "68" Micro Journal, May '83). Floats, etc.
FLEX, Color FLEX, OS/9 \$375.00 UniFLEX \$425.00

--- PASCAL ---

TSC PASCAL -- Native Code Compiler (UCSD Oriented). FLEX and Color FLEX \$200.00

Lucidata PASCAL -- P-Code Compiler (ISO Standard). Designed especially for Microcomputer Systems; Run-time System checks available resources for each task, allowing operation on even minimal computer systems. Allows linkage to Assembler Code for maximum flexibility.
FLEX and Color FLEX 5" \$190.00 FLEX 8" \$205.00

OmegaSoft PASCAL -- For the PROFESSIONAL; ISO Based, Native Code Compiler. For Real-Time and Process Control applications. Use custom I/O devices in place of the Pascal INPUT and OUTPUT; Long Int. (32 Bit); Dynamic length strings; Interrupt Processing, ROM-able, PIC, Re-Entrant Code, etc. POTENTIAL! Includes Source for the Symbolic Debugger, Runtime, and several Utilities. Requires a "Motorola Compatible" Relocating Assembler and Linking Loader.

#PCS2 FLEX and Color FLEX \$425.00 (one year Maint. \$100)

Add \$2.00 Per Item For Shipping & Handling

For Ordering Call TOLL
FREE 1-800-338-6800



FREE DISKETTE With Each \$50 Purchase

SOUTH EAST MEDIA

5900 Cassandra Smith Rd., Hixson, TN 37343

(615) 842-4601

SOUTH EAST MEDIA
 5900 Cassandra Smith Rd., Hixson, TN 37343
 (615) 842-4601

Software

For Ordering Call TOLL FREE 1-800-338-6800 **FLEX™ OS-9™ Color Computer**

— A full, screen oriented, WORD PROCESSOR —

STYLOGRAPH 2.0 — (now runs on the Data-Comp and FHL Color FLEX Systems; uses the 51 x 24 Display Screens). Full screen display and editing (i.e., what you see is what you get); supports the Daisy Wheel proportional printers.

SPECIAL Color FLEX STYLO \$195.00; FLEX and OS-9 STYLO \$295.00; UniFLEX STYLO \$395.00

Fast SPELLING CHECKER — allows directly changing the Text File, adding words to the dictionary, etc. 75,000 words in less than 400 sectors.

FLEX, Color FLEX, OS/9 \$125.00 UniFLEX \$175.00

MAIL MERGE — greatly extends the power and flexibility of STYLOGRAPH. Allows Multiple Text Files to be printed out as one large document. Provides for merging information into the Text File during printing (such as different names and addresses), etc.

FLEX, Color FLEX, OS-9 \$145.00 UniFLEX \$195.00

INFOMAG Data Base Management System -- An XBASIC-based, Menu Driven, DBMS with "Built-In" Audit Tracking, Extremely Powerful Report & Format Capabilities, etc. This Time Proven DBMS will become the "Work Horse" of your Software Stable.

FLEX and Color FLEX \$295.00 UniFLEX \$395.00

Accts Rec., Accts Payable & Gen Ledger — A FULL Accounting Package that can be used together, or as separate packages; provides the IRS required Audit Tracking. (XBASIC, based on the "Osborne Business Programs.")

FLEX and Color FLEX \$295.00/PROG UniFLEX \$395.00/PROG

An Electronic Spread Sheet

OTHCALC -- THE Electronic Spread Sheet for 6809 Computer Systems. An extremely POWERFUL Business Tool, this Program will find an unlimited number of "non-business" applications, also (for example, I have just finished setting up a Full Junior College Electronics Curriculum using OTHCALC). Advanced features like "Table Lookup" make Income Tax work easy; Column or Row Sorting for numerous applications; etc. Completely "Memory Resident", Machine Language, this Program is FAST. Provides STANDARD FLEX Text File output for use with BASIC, Word Processors, Pascal, "C", etc.

FLEX and SPECIAL Color FLEX (Both FHL and Data-Comp) \$200.00 UniFLEX \$395.00

Machine Language DATA BASE MANAGEMENT System

Westchester Applied Business Systems XIMS Data Management Systems. Possibly one of the most powerful DBMS's available, this machine language program is small enough to operate on a single sided 5" disk, yet provides the speed of M.L. and power limited only by the user's imagination. Supports Sequential, Hierarchical, and Random Access File Structures, and has Virtual Memory capabilities for those Giant Data Bases. Easy-to-use English Language Command Structure.

XIMS -- FLEX and Color FLEX \$179.95

XIMS + -- FLEX and Color FLEX \$250.00

UNIVERSAL DATA RESEARCH INC. — Note: ALL Accounting and DBM Progs. Require FLEX and XBASIC. These are Time Tested programs from an old, established, software house; for Color FLEX Systems

Data Base Manager Part 1 - \$49.95; Data Base Manager Part 2 - \$49.95

Church Contributions - \$49.95 Single Entry Gen Ledger - \$49.95 Balanced Billing System - \$49.95

Integrated Software for Color FLEX

A/C \$99.95

A/P \$99.95

Gen Ledger \$189.00

Inventory 2 \$69.00

Payroll \$99.95

FLEX and UniFLEX — Note: Requires XBASIC (FLEX) or basic (UniFLEX)

A/C - FLEX \$295

UniFLEX \$395

A/R - FLEX \$295

UniFLEX \$395

Gen Ledger - FLEX \$295

UniFLEX \$395

Inventory 2 - FLEX \$295

UniFLEX \$395

Payroll - FLEX \$295

UniFLEX \$395

DBM - FLEX \$350

UniFLEX \$450

Please specify 5 or 8 inch disk when ordering all software!

Computer Systems Consultants FLEX XBASIC Programs

FULL SCREEN FORMS DISPLAY

FLEX and Color FLEX \$50.00

UniFLEX \$75.00

FULL SCREEN MAILING LIST

FLEX and Color FLEX \$100.00

UniFLEX \$110.00

FULL SCREEN INVENTORY/MRP

FLEX and Color FLEX \$100.00

UniFLEX \$150.00

TABLE BASE SPREADSHEET

FLEX and Color FLEX \$100.00

UniFLEX \$200.00

*FLEX is a trademark of Technical Systems Consultants
 *OS9 is a trademark of Microware

Add \$2.00 Per Item For Shipping & Handling

For Ordering Call TOLL
 FREE 1-800-338-6800



FREE DISKETTE With Each \$50 Purchase

SOUTH EAST MEDIA

5900 Cassandra Smith Rd., Hixson, TN 37343

(615) 842-4601

SOUTH EAST MEDIA
5900 Cassandra Smith Rd., Hixson, TN 37343
(615) 842-4601

Software

For Ordering Call TOLL FREE 1-800-338-6800 **FLEX™ OS-9™ Color Computer**

SPELLB "Computer Dictionary" — OVER 120,000 words!

No more "Let your fingers do the walking through the dictionary" while you are inputting Text with your favorite Editor or Word Processor. **SPELLB** is more than "another Spelling Checker"; it allows you to look up a word from within your Editor or Word Processor so that you **KNOW** it is right **WHEN YOU TYPE IT IN** with the **SPB.CMD** Utility (which operates in the **FLEX** Utility Space). Yes, it **ALSO** allows you to check and update the Text after you are finished; along with allowing you to **ADD WORDS** to the Dictionary, "Flag" questionable words in the Text for evaluation later, "View a word in context" before changing or ignoring, etc. **SPELLB** first checks a "Common Word Dictionary", then the normal Dictionary, then a "Personal Word List", and finally, any "Special Word List" you may have specified. **SPELLB** also allows the use of **Small Disk Storage** systems.

FLEX and Color FLEX \$129.95

JUST — a Text Formatter

JUST, a Text Formatter developed by Ron Anderson, provides numerous features which make it a valuable addition to any **FLEX** Users Software Library. **JUST** is designed for formatting Text Output for **Dot Matrix Printers** and provides many unique features:

- Output the "Formatted" Text to the Display for format analysis and change.
- Output the "Formatted" Text to a Text File for use with the supplied **PPRINT.CMD** for producing multiple copies of the Text on the Printer **INCLUDING IMBEDDED PRINTER COMMANDS** (this Utility is very useful at other times also, and worth the price of the program by itself).
- "User Configurable"** for adapting to other Printers (comes set up for **Epson MX-80** with **Grafrax**); provides for up to ten (10) imbedded "Printer Control Commands", such as **Italics on and off**, **boldface on and off**, etc.
- Automatic compensation for a "Double Width" printed line.
- Includes the normal line width, margin, indent, paragraph, space, vertical skip lines, page length, page numbering, centering, fill, justification, etc.
- Use with **ANY** Editor.
- Supplied with "Structured Source" (Winirush PL/9); easy to see the flow of the program.

FLEX and Color FLEX \$49.95

SPECIAL! SPECIAL! SPECIAL!

Star-Kits excellent **SPELL 'N FIX** Dictionary and **WRITE 'N SPELL** Word Look Up program **IN ONE PACKAGE**;

FLEX and Color FLEX Systems — BOTH for ONLY \$150.00

When these are gone; the price goes UP!! WAY UP!! **ORDER NOW!!**

Also, call for "More Info" on both the **FLEX** Based and **Color Computer** Based **STAR-Kits** Products; including the **HUMBUG** Monitor, Check 'N Tax Program, **REMO TERM** Color Computer External Terminal Program, etc.

PASCAL UTILITIES — Requires LUCIDATA Pascal ver 3.

INDEX — produce a Cross Reference Listing of any text; oriented to Pascal Source.

INCLUDE — allows the inclusion of other Files in a Source Text; has unlimited nesting capabilities. Also allows Binary File inclusions.

PROFILER — produces an Indented, Numbered, "Structogram" of a Pascal Source Text File. Allows viewing the overall structure of large programs, and provides clues as to the integrity of the program. Supplied as Source Code; requires compilation.

FLEX and Color FLEX — Each program \$25.00

COPYCAT — (Pascal **NOT** required) Allows reading **TSC Mini-FLEX**, **SSB 00668**, and **Digital Research CP/M** Disks while operating under **FLEX 1.0**, **FLEX 2.0**, or **FLEX 9.0** with **6800** or **6809** Systems. **COPYCAT** will not perform Miracles, but, between the program and the manual, you stand a good chance of accomplishing a transfer. Includes Utilities to List Directories, Copy Files, and convert Text Files when required. Also includes a Utility for investigating Physical Compatibility problems. Programs supplied in **Modular Source Code** to make it easier to solve unusual problems.

FLEX and Color FLEX 5" \$50.00 FLEX 8" \$65.00

*FLEX is a trademark of Technical Systems Consultants
**OS9 is a trademark of Microware

Add \$2.00 Per Item For Shipping & Handling

For Ordering Call TOLL
FREE 1-800-338-6800

SOUTH EAST MEDIA



5900 Cassandra Smith Rd., Hixson, TN 37343

FREE DISKETTE With Each \$50 Purchase

(615) 842-4601

SOUTH EAST MEDIA
5900 Cassandra Smith Rd., Hixson, TN 37343
(615) 842-4601

Software

For Ordering Call TOLL FREE 1-800-338-6800 **FLEX™ OS-9™ Color Computer**

O-F — OS/9 to FLEX - FLEX to OS/9 —

Finally, the barrier has been removed between OS/9 and FLEX formatted disks! Now you can READ from, and WRITE to, a Single Sided 5" or 8" FLEX diskette from OS-9 with O-F. O-F is a new and unique program, written in BASIC9 (with Source), that performs the following functions:

REFORMAT: A BASIC9 Program that reformats a chosen amount of an OS-9 disk to FLEX Format so it can be used normally by FLEX.

FLEX: A BASIC9 Program that does the actual read or write function to the special O-F Transfer Disk, all selectable from a user-friendly menu. Functions provided include reading the FLEX Directory, Deleting FLEX Files, Copying both directions, etc. All selections are interactive and complete, including all necessary prompts to the operator.

FLEX users can read, write and use the special disk as any other FLEX disk, provided the FLEX directory is not allowed to continue beyond track zero (too many files).

FLEX and Color FLEX \$79.95

COPTMULT.CMD — Copy LARGE Disks to several smaller disks —

The following FLEX utilities allow the backup of ANY size disk to any SMALLER size diskettes (Winchester to 8's or 5's, 8" to 5's, etc.). By simply inserting diskettes as requested by COPTMULT, a large disk system may be downloaded to your present floppy disk system, any size. No need to fiddle with directory deletions or any of the other tedious operations that must be done using the normal copy routines.

COPTMULT.CMD understands normal "copy" syntax and always keeps up with files already copied by maintaining directories for both host and receiving disk system, eliminating hours of tedious keyboard entries and other time consuming cleanup chores.

BACKUP.CMD is a special program that downloads "random" type files, any size.

RESTORE.CMD a special program to restructure copied "random" files for copying, or recopying back to the host system.

FREELINK.CMD a "bonus" utility that "relinks" the free chain of floppy or hard disk thereby eliminating fragmentation.

Completely documented source files included. ALL 4 Programs \$99.50 (8" or 5")

CHESS 6809

Requires FLEX and DISPLAYS On Any Type Terminal

Features:

- *Two display boards. *Change skill level. *Swap side. *Point scoring system.
- *Four levels of play. *Solve Checkmate problems in 1-2-3-4 moves.
- *Make move and swap sides. *Play white or black.

This is one of the strongest CHESS programs running on any microcomputer, estimated USCF Rating 1600+ (better than most 'club' players at higher levels).

FLEX and Color FLEX \$79.95

DIET-TRAC Forecaster

DIET-TRAC Forecaster is an X BASIC program that plans a diet in terms of either calories and percentage of carbohydrates, proteins and fats (C P G%) or grams of Carbohydrate. Protein and Fat food exchanges of each of the six basic food groups (vegetable, bread, meat, skim milk, fruit and fat) for a specific individual.

Sex, Age, Height, Present Weight, Frame Size, Activity Level and Basal Metabolic Rate for normal individual are taken into account. Ideal weight and sustaining calories for any weight of the above individual are calculated. When a weight goal is given (either gain or loss), and a calorie plan is agreed upon between the computer and the individual, the number of days to reach the weight goal is projected. The starting and ending rate of weight loss is calculated, and a daily calendar with each day's weight for a 30-day period is printed.

FLEX - \$59.95 UniFLEX - \$89.95

XDATA — A COMMUNICATIONS Package for the UniFLEX Operating System —

Allows UniFLEX Based Systems to Transmit and Receive files to and from other Computer Systems via Modem. Use with CP/M, Main Frames, other UniFLEX Systems, etc.

- Verifies Transmission integrity using checksum or CRC
- Automatically Re-Transmits bad blocks
- Transmits data in 128 byte blocks

UniFLEX \$299.99

*FLEX is a trademark of Technical Systems Consultants
*OS9 is a trademark of Microsoft

Add \$2.00 Per Item For Shipping & Handling

For Ordering Call TOLL
FREE 1-800-338-6800



FREE DISKETTE With Each \$50 Purchase

SOUTH EAST MEDIA

5900 Cassandra Smith Rd., Hixson, TN 37343

(615) 842-4601

SOUTH EAST MEDIA
5900 Cassandra Smith Rd., Hixson, TN 37343
(615) 842-4601

Software

For Ordering Call TOLL FREE 1-800-338-6800 FLEX™ OS-9™ Color Computer

AT LAST!! A FULL BLOWN DISASSEMBLER FOR THE COLOR COMPUTER

Computer Systems Consultants **SUPER SLEUTH** is a "Time Tested", reliable, **PROVEN** Disassembler that has gained acceptance through out the FLEX Community as an extremely **POWERFUL, INTERACTIVE** Software Tool. Now, this powerful Disassembler has been converted to run on a **Standard 32K Color Computer** or **TDP-100 System with a Disk System**. The **CoCo SLEUTH** Software Package consists of 3 Programs; **SLEUTH** (the Disassembler), **CHGNAM** (used to globally Change Labels to a meaningful Name), and **XREF** (a Cross Reference Generator for Source Code Files). **CoCo SLEUTH** will Disassemble Disk Files of 6800, 6801, 6802, 6803 (the "Baby CoCo"), 6805, 6808, 6809, and 6502 (Apple, Atari, Commodore, etc.) Object Code if you can get it on a Color Computer Disk. (See Aug. '83 '68' Micro Journal "Color Users Notes" Column for a full Review.)

Color Computer Disk - Object Code Only \$49.00

FORTH Programming Language

Stearns Electronics FORTH -- Intrigued by **Forth??** Here is a **Forth** package tailored to the **Color Computer**! This package is supplied on Tape, with instructions for transferring it to disk if you wish. Written primarily in machine language, it's **speed is unparalleled**. A full **Semigraphic-8 Editor** is provided, along with "goodies" like **Graphics** and **Sound Commands**, **Printer Commands**, **Auto-Repeat** and **Control Keys**, etc. If you are interested in **Learning Forth**, a **Trace Feature** is provided which is invaluable. If you are a **FORTH Pro**, this package provides **CPU carry flag accessibility**, **Fast Task Multiplexing**, **Clean Interrupt Handling**, etc. (Or; you won't "out grow" the basic capabilities of this Implementation). Combine this package with **Leo Brodie's EXCELLENT** book "Starting FORTH", and you will be a **FORTH Expert** before you know it (and have a lot of fun doing it!).

Color Computer TAPE (w/ instructions for transferring to Disk) \$58.95

Color Computer GRAPHIC SCREEN PRINT Programs

Dumps any "PMODE" Screen to the Printer with the **BASIC USR** function. Shift the Printout Left or Right or **Reverse Print** (Dark for Light Screen and Vice Versa). All Programs on Tape.

GSFR for Radio Shack LP-VII/VIII & DMP 100/200/400 Printers	\$7.95
GSPE for Epson w/ Graftrax and Graftrax + Printers	9.95
GSFG for Gemini 10 and 15 Printers	9.95
GSPP for the Prowriter Printers	9.95

DATE-O-BASE CALENDAR Program

A Menu Driven **EXTENDED BASIC** Program which allows the entry of up to 12 Memos per Day, each of which may contain up to 28 Characters, for any day of the Month between the years 1700 and 2099. A **Graphic Calendar** shows which days contain Memos, and a "Key Word" Search is provided which can be output to the Screen or Printer.

TAPE DATE-O-BASE CALENDAR (Each Tape File will hold up to 400 Memos)	\$16.95
DISK DATE-O-BASE CALENDAR (4,000 Memos at 300/Month per Disk)	19.95

Interested in INTEREST (the Money Kind)?

An **EXTENDED BASIC** Program that will help you deal with numerous problems requiring interest calculations. Present Value, Rate of Return, Current Bond Yield and Rate of Return to maturity, Loan Repayment Amortization Schedules, etc.

TAPE \$29.95

Data Base Management System

DISK DATA MANAGER 64K - EXTENDED BASIC w/ Mach. Lang. Routines. Allows a max of 246 Chars. and 14 Fields per Record, and another Record can be linked to the first; 8 Char. Field Names, up to 99 Chars. per Field. Powerful On-Screen editor for input and update, flexible Output capabilities including output to Disk Files for use by other Programs. Change File Definition without re-entering the Data, Split Files, etc. Allows Multiple Field Sorts, Select on any combination of Fields, etc. An extremely **POWERFUL TOOL**; instructions provide examples of Mailing Lists and a financial Stock Profit and Loss Tracking System.

DISK \$54.95

ACCOUNTING

DISK DOUBLE ENTRY - DISK EXTENDED BASIC w/ Mach. Lang. Routines. A "Traditional" Accounting Package for Small Business, Clubs, Churches, Personal Use, etc. Up to four levels of subtotals with Trial Balance, Income Statement, and Balance Sheet Reports. DDE allows up to 300 accounts and a Trial Balance of \$9,999,999.99. Transactions may be up to 14 lines long, and comments and explanations may be freely used. Accounts are traceable to the journal transaction, which may include comments. Screen reports allow review of past transactions and current balances.

DISK \$44.95

*FLEX is a trademark of Technical Systems Consultants
*OS9 is a trademark of Microware

For Ordering Call TOLL
FREE 1-800-338-6800



Add \$2.00 Per Item For Shipping & Handling

FREE DISKETTE With Each \$50 Purchase

SOUTH EAST MEDIA

5900 Cassandra Smith Rd., Hixson, TN 37343

(615) 842-4601

SOUTH EAST MEDIA
5900 Cassandra Smith Rd., Hixson, TN 37343
(615) 842-4601

Software

For Ordering Call TOLL FREE 1-800-338-6800 **FLEX™ OS-9™ Color Computer**

DYNASHARE --- Multi-User, Multi-Tasking with **FLEX**

Southeast Media is now shipping **DYNASHARE FROM STOCK** - the multi-user, multi-tasking capability of **DYNASHARE** allows **FLEX** users the advantages of more sophisticated and time saving computer usage without having to buy or learn a new Language or Operating System syntax. **DYNASHARE**, as its name implies, allows true "time-sharing" operation under the popular **FLEX** operating system, and also allows each user to run two simultaneous jobs (multi-tasking); even on single-user systems. For example, while in **EDIT**, you can list another file or examine a directory. Or, you might look up an item in a Data Base while a Sort is in progress! **DYNASHARE** also provides some fringe benefits that will be greatly appreciated by **FLEX** users, including type-ahead, command line editing, and instant response to "escape".

DYNASHARE is the painless method! Use your existing Flex computer by simply adding 64K of RAM for each user. Fact is, you still use **FLEX** just like you always have! **DYNASHARE** is not intended as competition to **UnifLEX**. It does not improve on the speed of **FLEX**, and does not offer password protection or other niceties of a full-blown multi-user system. What **DYNASHARE** does do is give **FLEX** users a low-cost way to use existing software in a multi-user, multi-tasking environment, so your existing **FLEX** versions of **BASIC**, **XBASIC**, editors, assemblers, disassemblers, sort/merge packages, word processors, compilers, **DYNACALC** spread-sheet package, and so on are still good.

NOTE -- The initial release of **DYNASHARE** is for **SWTC S/09** Computers, but versions will also be available for other popular extended-memory (up to 1024K) systems, such as **HELIX** and **GMDX**. A minimum of 128K of RAM will be required with ALL versions. **DYNASHARE** requires 64K of RAM for each active task; thus a 256K system could allow foreground-background operation on two terminals, or foreground-only operation on four terminals.

AVAILABLE NOW from **SOUTHEAST MEDIA** --- \$200.00

--- AUTHORS - PROGRAMERS ----- QUALITY SOFTWARE NEEDED ---
FLEX - UnifLEX - OS/9 - Color Computer

For the past several months, we at the **Southeast Media Division of Computer Publishing, Inc. (CPI)**, the parent company of **'68' MICRO JOURNAL** and **COLOR MICRO JOURNAL**, have debated expanding our software distribution business. Many other magazines have been doing so for years (in fact, MOST were in the Software Distribution Business BEFORE they began to publish a Magazine). Presently there are many fine examples of software that has been developed by YOU, our readers, that will never see the "light of day" due to the cost of Advertising and TIME and cost involved in the production, distribution, and Customer SUPPORT of that software unless SOMEONE, with enough exposure and the willingness to continually advertise, runs with the ball.

Software is the "backbone" for the REAL utilization of any Computer System, and ours are no exception! This has been no simple decision. While we realize that there could be some conflict with some of our advertisers, we ALSO hear a LOUD and CONTINUOUS cry for HELP from our Readers. From day one, the foremost concern of **'68' MICRO JOURNAL** has been it's READERS! Therefore, our **Southeast Media Division** will accept, for appraisal for possible Distribution, 6809 software; Games, Utilities, Software Development, Business Application Programs, etc.

In the past there has been too much software offered that was not quite ready. We will strive to eliminate that element. But, right up front, we tell you only that we will do our very best; nothing more. Also, we will strive to keep cost to a bare minimum, while securing for the author a fair return in royalty payments, promptly paid, and in customer support for his product.

Of course, we will expect, no -- **DEMAND**, that the author keep the product free of errors (bugs), and maintain it in a prompt and business like manner. Also we shall require that authors be willing to furnish 'source' for those programs that justify, by price and utility, inclusion of same. The lack of source code, properly commented, is a continual complaint we hear. Not all programs will be sold with source, but where necessary, we will insist that it be included.

In some instances the program may be small or short and not justify itself as a "single" sale product. In this event it will be combined with other like programs, and offered as a package. In that event, the royalties will be split between the various authors.

If you have software that you feel will qualify under this program, please contact one of the people below. Remember, if your software has any problems or "funnies" -- **GET IT STRAIGHT BEFORE YOU CONTACT US!!** Also get your source code in proper shape and well commented; there is too much 99% code already drifting around.

If your software is **READY** contact: Bob Bay, Don Williams, or Tom Williams

Southeast Media is a division of **Computer Publishing, Inc. (CPI)**,
a family of 100% **6809** support facilities.

*FLEX is a trademark of Technical Systems Consultants

**OS9 is a trademark of Microware

For Ordering Call TOLL
FREE 1-800-338-6800



Add \$2.00 Per Item For Shipping & Handling

FREE DISKETTE With Each \$50 Purchase

SOUTH EAST MEDIA

5900 Cassandra Smith Rd., Hixson, TN 37343

(615) 842-4601

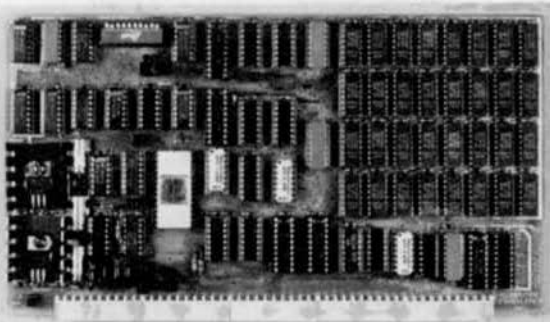
COMPUTER EXCELLENCE, INC.

IS PROUD TO PRESENT A SUPERB

256K MEMORY SYSTEM

FOR THE SS-50 OR SS-50c BUS.

Now any 6800 or 6809 system can have up to a MEGABYTE using our sophisticated new 256K memory system.



Runs full speed: 2MHz, no MADDY, no interrupts and still executes a refresh cycle EVERY system 'E' clock. Works with any processor board, with or without 'DATs'. High-speed DATs on board, standard. Works with any SS-50 bus, with or without extended addressing lines. No cables or bus modifications. Board is QUIET: over 6 ul of ceramic decoupling capacitance. All gold bus connectors, fully socketed, solder masked. Does not use the double height sockets. Sockets are selective gold plated for reliability. Assembled, tested, ready to go. Ready to work with the 256K D-RAMS when they become available. Use all the memory, including \$E000-\$F000: no useless banks. Memory management scheme compatible with SWTPC and GIMIX. Virtual disk drivers, memory diagnostics supplied with board. Software is on 5 inch, SS, SD, 35 track disk for FLEX. Provides up to 3,750 sectors of virtual disk storage at RAM speed. Requires less than 1.2 Amps from your systems +8 volt bus. Many configuration options.

And it's ready NOW! Shipped from stock!

Introductory price: **\$749.00**

Including software, users and maintenance manuals or send \$15.00 for users manual, applicable toward purchase price. COD. Cashier's check, personal checks must clear before shipment. Florida residents add 5% sales tax. 4834 N.E. 12 AVE., FT. LAUDERDALE, FL 33334(305) 762-8321

*FLEX is a trademark of Technical Systems Consultants

CLASSIFIED ADVERTISING

CONTINUED FROM PAGE 49

F&D MDI-1 Disk Controller with EXS-1 Data Separator, two Siemens FDD100-5 Disk Drives (Floppy) with case, PS, Cables & Software all for \$550. Bob Phelps (606) 236 9400

*** UNIFLEX Users - SAVE \$\$\$\$ ***

New, never installed or used - High Speed SWTPC A-1 Tape Back-up System, 20 megabyte capacity - Complete with new DMF3 Disk Controller - Save over \$1,000.00 - **ONLY \$2475** Don't get caught without that BACKUP for all your data **DON'T MISS THIS ONE!!** - Ask for Don, Tom or Larry, Days

A/C 800 338-6800.

LIKE NEW - ADDS-VIEWPOINT CRT TERMINALS - DEMO UNITS, THESE ARE LIKE NEW - WARRANTY NEVER REGISTERED - SAVE - **ONLY \$478.00 EACH** - ONLY 2 LEFT. CALL ASK FOR DON, TOM OR LARRY.

A/C 800 338-6800.

GIMIX 11 CPU+, 2 Mhz, with FLEX and OS-9 - Level 1, Ver. 1.2, latest as of 9/1/83. This CPU board is like new, we went to GIMIX 111 system. Set up and ready to run. With FLEX and OS-9. Have both on your system and just call FLEX or OS-9. Includes software and documentation. Also Time of day clock, FLEX DAT, OS-9 FPLA. Set up to require GIMIX #68 DMA disk controller. Original cost over \$900.00. Special only \$729.95 plus shipping (\$3.50). Ask for Don, Tom or Bob - 1 615 842-4601. This will not last long!

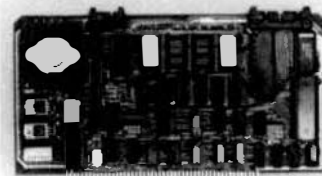
TELETYPE Model 43 PRINTER - with serial (RS232) Interface, and full ASCII keyboard. LIKE NEW - New cost \$1295.00 - **ONLY \$759.00** ready to run - Call Tom - Larry - Bob, CPI 615 842-4600.

6809 + OS-9* + S-100

WHAT A COMBINATION!

ADS is pleased to announce support for Microware's OS-9 on the ADS 6809 S.B.C.

Used in Conjunction with our Octafloppy™ disk controller card, you now have complete 6809/OS-9 capabilities for the S-100 bus. Further, the ADS Octafloppy™ gives you up to eight drives, 5 1/4" or 8",



single or double sided, single or double density. Expand to the power of the S-100 bus. (The bus that handles 16 bit CPU's too!) Call or write today for more information.

ads

Ackerman Digital Systems, Inc.

110 N. York, Suite 208 • Elmhurst, IL 60126
(312) 530-8992

See us at
Microware OS9 Seminar!

OS-9* Trademark of Microware Systems Corp.

*** ALFORD'S GOOD NEWS ***

PRE-INVENTORY CLEARANCE SALE!!

It's inventory time around here. To make things easier on ourselves, we want to sell as much of our current inventory as we can. So until stock is depleted, we are offering some FANTASTIC DEALS on selected items. Remember, these prices are ONLY good until our current stock is gone, so call or write to place your order today!

VS-1 SPEECH SYNTHESIZER.....\$120

This is the most popular unlimited speech synthesizer available for the SS-50 I/O bus! All units are completely assembled and tested, LESS the SC-01A synthesizer I.C. If you buy this unit while on sale, we will include a copy of our SC-1 text translator program at no additional charge. This gives you a \$170 value. You save \$50!! Be sure to specify 6800 FLEX, 6809 FLEX, OS-9, DOS68 or DOS69 operating system and disk size when ordering.

SP-1 SPEECH SYNTHESIZER.....\$110

Our popular unlimited speech synthesizer for the Radio Shack Color Computer is also available on this sale. All units are completely assembled and tested, LESS the SC-01A synthesizer I.C. Buy your SP-1 on this sale and we will INCLUDE our SC-1/COLOR translator program which lets your CoCo speak English using nothing more than PRINT statements! Also, we will throw in our COLOR-TREK+ program. This is our COLOR-TREK program adapted for the SP-1. English messages are continually spoken during game play! The action is non-stop. This makes a \$170+ value. You save 50%! Software is available on CASSETTE ONLY.

SC-01R SYNTHESIZER IC'S.....\$ 55

These chips are the ones used in our synthesizer boards, and are prime, new parts directly from Votrax. We have a limited number of this most popular speech synthesizer I.C. in stock for this sale. Suggested retail price is \$85. Discount houses are selling them for \$70. You save \$15 on the discount house price. We include copies of the Votrax Spec sheets with these parts.

SCREDITOR-III.....\$150

All of our customers who have used both of the other major word processing packages available for 68xx systems tell us that SCREDITOR III is the best program that they have ever seen (in truth, ONE customer liked one of the others better, but you can't please EVERYONE!). As of July 15th, SCREDITOR III will cost \$175. If you have been considering a word processor, or just need a better editor than what you are using now, this is the time to buy SCREDITOR III! For more information about this fantastic program which includes over 150 pages of documentation (and another 240 pages coming soon!), see our ads in the past several issues of Micro-Journal. Be sure to specify 6800 FLEX, 6809 FLEX, OS-9, DOS68 or DOS69 operating systems and disk size when ordering.

ORDERING INFORMATION...

All sales are FINAL. No returns or refunds, NO open account orders and NO DEALER DISCOUNTS on VS-1, SP-1 or SC-01A purchases for the duration of this sale. Normal dealer discounts and open account terms will apply to SCREDITOR III purchases.

All overseas orders MUST include \$10 for shipping. All North American orders under \$100 MUST include \$5 for shipping. ALL checks MUST be drawn in U.S. dollars on accounts in U.S. banks. No foreign banks or currencies will be accepted under ANY conditions. All Virginia orders MUST include 4% state sales tax. COD orders in the U.S. ONLY, and must be shipped by UPS. All overseas orders shipped by AIR PARCEL POST. We gladly accept VISA and MASTERCARD orders.

Please allow two to three weeks for order processing. Orders paid by check may take longer. Stocks are limited. Any orders received after stocks are exhausted will be returned unprocessed.

ALFORD AND ASSOCIATES
PO BOX 6683
RICHMOND VA 23230
PH 804-320-6722

FLEX is a trademark of Technical Systems Consultants. OS-9 is a trademark of Microware Systems Corporation.

The Lord is our primary associate. Too often with the pressures of business today, time is not taken to acknowledge His presence, and too often we fail, due to our human weaknesses, to show His influence in our lives. We thank each of our customers for their patience and prayers for us when we do not reflect Christ in our business activities.

CHRISTMAS SALE

Make a loved one happy by giving yourself a Star-Kits program for Christmas. It's easy with these special prices until December 24th.

SPELL 'N FIX for 6809 Flex, normally \$178.58, on sale for \$125. Other SS-50 versions, normally \$89.29, now \$75. (And if you didn't know, Flex is a trademark of Technical Systems Consultants, who asked us to tell you.)

WRITE 'N SPELL, normally \$75.11, now \$60 alone or \$30 when ordered with **SPELL 'N FIX**.

HUMBUG ON DISK. This fantastic debugger runs as a utility under either Flex or STAR-DOS 64, especially useful for 6809 Flex on the Color Computer! Regularly \$59.95, now \$49.95.

HUMBUG in ROM for SS-50 systems reduced \$10; send for spec sheet for versions available.

STAR-DOS 64 disk operating system for the Color Computer, normally \$74.90 now \$49.90.

We accept checks, COD, MasterCard, or Visa. NY Residents please add appropriate tax. Have a Merry Bah Humbug.

STAR KITS

P. O. Box 209 - M
MT. KISCO, N.Y. 10549
(914) 241-0287

OS9 APPLICATION SOFTWARE

ACCOUNTS
PAYABLE

\$299

ACCOUNTS
RECEIVABLE

\$299

GENERAL
LEDGER
with
CASH

JOURNAL

\$399

PAYROLL

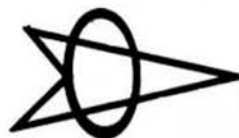
\$499

SMALL
BUSINESS
INVENTORY

\$299

COMPLETE DOCUMENTATION \$19.95

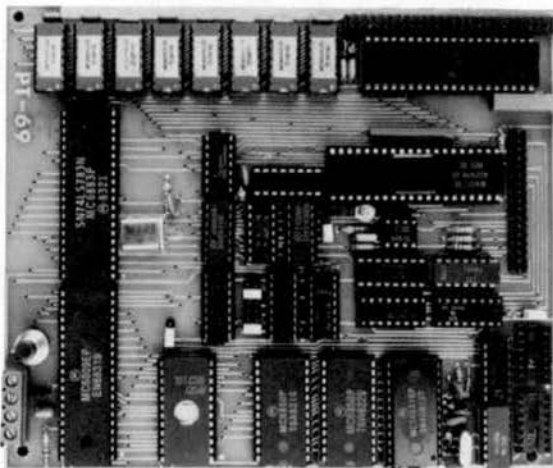
OS9 & BASIC 09 ARE TRADEMARK OF
MICROWARE, INC. & MOTOROLA CORP.



**SPECIALTY
ELECTRONICS**

(405) 233-5564
2110 W. WILLOW - ENID, OK 73701

6809 SINGLE BOARD COMPUTER 6809 ADVANCED TECHNOLOGY ECONOMICAL



BOARD SIZE
5½" x 6½"

The new PT-69 is a complete computer system on a single card. Compact, powerful, with performance equal to bigger systems. The PT-69 features:

- 6809E Processor
- Double Sided/Double Density Floppy Disk Controller
- 2 RS232 Serial Ports (6850)
- 2 8-Bit Parallel Ports (6821)
- Time-of-Day Clock (MC146818)
- Power Requirements are +5 @ 800MA and ±12 @ 20MA
- Run FLEX 9 or SWTPC FLEX
- 56K RAM: 4K EPROM: 4K I/O
- LOW COST:
 - * PT-69, Assembled and Tested \$295.95
 - * PT-69, Assembled, Tested, with Power Supply and Cabinet \$374.95
 - * COMPLETE SYSTEM with PT-69 Board, 2 DS/DD 5¼" Drives, Cabinet, and Power Supply \$895.95

— Kits are available -
Write for Details!

PERIPHERAL TECHNOLOGY

"Supplying Your Computer Needs Since 1978"

3760 Lower Roswell Road
Marietta, Georgia 30067

VISA/MASTERCARD/CHECK/COD 404/973-0042

™ FLEX is a trademark of Technical Systems Consultants

TEN MOST-ASKED QUESTIONS about **DYNACALC**TM

THE ELECTRONIC SPREAD-SHEET FOR 6809 COMPUTERS

- 1. What is an electronic spread-sheet, anyway?**
Business people use spread-sheets to organize columns and rows of figures. DYNACALC simulates the operation of a spread-sheet without the mess of paper and pencil. Of course, corrections and changes are a snap. Changing any entered value causes the whole spread-sheet to be re-calculated based on the new constants. This means that you can play, 'what if?' to your heart's content.
- 2. Is DYNACALC just for accountants, then?**
Not at all. DYNACALC can be used for just about any type of job. Not only numbers, but alphanumeric messages can be handled. Engineers and other technical users will love DYNACALC's sixteen-digit math and built-in scientific functions. You can build worksheets as large as 256 columns or 256 rows. There's even a built-in sort command, so you can use DYNACALC to manage small data bases — up to 256 records.
- 3. What will DYNACALC do for ME?**
That's a good question. Basically the answer is that DYNACALC will let your computer do just about anything you can imagine. Ask your friends who have VisiCalcTM, or a similar program, just how useful an electronic spread-sheet program can be for all types of household, business, engineering, and scientific applications. Typical uses include financial planning and budgeting, sales records, bills of material, depreciation schedules, student grade records, job costing, income tax preparation, checkbook balancing, parts inventories, and payroll. But there is no limit to what YOU can do with DYNACALC.
- 4. Do I have to learn computer programming?**
NO! DYNACALC is designed to be used by non-programmers, but even a Ph.D. in Computer Science can understand it. Even experienced programmers can get jobs done many times faster with DYNACALC, compared to conventional programming. Built-in HELP messages are provided for quick reference to operating instructions.
- 5. Do I have to modify my system to use DYNACALC?**
Nope. DYNACALC uses any standard 6809 configuration, so you don't have to spend money on another CPU board or waste time learning another operating system.
- 6. Will DYNACALC read my existing data files?**
You bet! DYNACALC has a beautifully simple method of reading and writing data files, so you can communicate both ways with other programs on your system, such as the Text Editor, Text Processor, Sort/Merge, STYLOGRAPHTM word processor, RMSTM data base system, or other programs written in BASIC, C, PASCAL, FORTRAN, and so on.
- 7. How fast is DYNACALC?**
Very. Except for a few seldom-used commands, DYNACALC is memory-resident, so there is little disk I/O to slow things down. The whole data array (worksheet) is in memory, so access to any point is instantaneous. DYNACALC is 100% 6809 machine code for blistering speed.
- 8. Is there a version of DYNACALC for MY system?**
Probably. You need a 6809 computer (32k minimum) with FLEXTM, UNIFLEXTM, or OS-9TM operating system. You also need a decent crt terminal, one with at least 80 characters per line, and direct cursor addressing. If your terminal isn't smart enough for DYNACALC, you probably need a new one anyway. The UNIFLEX and OS-9 versions of DYNACALC allow you to mix different brands of terminal on the same system. There's also a special version of DYNACALC for Color Computers equipped with FLEX (Frank Hogg or Data-Comp versions).
- 9. How much does DYNACALC cost?**
The FLEX versions are just \$200 per copy; UNIFLEX version \$395; OS-9 version (works with LEVEL ONE or LEVEL TWO) \$250. Orders outside North America add \$7 per copy for postage. We encourage dealers to handle DYNACALC, since it's a product that sells instantly upon demonstration. Call or write on your company letterhead for more information.
- 10. Where do I order DYNACALC?**
See your local DYNACALC dealer, or order directly from CSC at the address below. We accept telephone orders from 10 am to 6 pm, Monday through Friday. Call us at 314-576-5020. Your VISA or MasterCard is welcome. Please specify diskette size for FLEX or OS-9 versions. Software serial number is required for the UNIFLEX version.

Order your **DYNACALC** today!

Foreign Dealers:

Australia & Southeast Asia: order from Paris Radio Electronics, 161 Bunnerong Road (PO Box 380) Kingsford, 2032 NSW Australia. Telephone: 02-344-9111.

United Kingdom: order from Compusense, Ltd., PO Box 169, London N13 4HT. Telephone: 01-882-0681.

Scandinavia: order from Swedish Electronics hk AB, Murargatan 23-25, Uppsala S-754 37 Sweden. Telephone: 18-25-30-00.

Computer Systems Center
13461 Olive Blvd.
Chesterfield, MO 63017
(314) 576-5020



UNIFLEX software prices include maintenance for the first year.

DYNACALC is a trademark of
Computer Systems Center

VisiCalc is a trademark of VisiCorp.
STYLOGRAPH is a trademark of Great Plains Computer Co.
RMS is a trademark of Washington Computer Services.
FLEX and UNIFLEX are trademarks of TSC.
OS-9 is a trademark of Microware and Motorola.

COMPARE

our EPROM PROGRAMMER with the field.

All data taken directly from manufacturer's current advertising. Software, interface, or Personality modules may also be required at additional cost.

- Triple voltage EPROM
- Supplied in kit form

		A	B	C	D	E	F
INTERFACE	S30	PAR	PAR	SER	S30	SER	SER
INTELLIGENT	NO	NO	NO	YES	NO	YES	YES
PROGRAMS							
2704*	•		•		•	•	•
2508	•		•	•	•	•	•
2708*	•	•	•	•	•	•	•
2758	•	•	•	•	•	•	•
2518	•	•	•	•	•	•	•
2718	•	•	•	•	•	•	•
2718*	•	•	•	•	•	•	•
2532	•	•	•	•	•	•	•
2732	•	•	•	•	•	•	•
2732A	•	•	•	•	•	•	•
2584	•	•	•	•	•	•	•
2784	•	•	•	•	•	•	•
2528	•	•	•	•	•	•	•
2728	•	•	•	•	•	•	•
2818	•	•	•	•	•	•	•
88784	•	•	•	•	•	•	•
8748	•	•	•	•	•	•	•
8749	•	•	•	•	•	•	•
TOTAL	11	3	12	6	11	11	11
PRICE	\$125	\$45	\$169	\$289	\$375	\$489	\$575

EPROM Programmer, \$125. Personality module for 2508, 2758, 2516, and 2716 included. Specify CPU, disk size, and operating system (TSC's FLEX or SSB's DOS) when ordering. Manual only, \$10; refundable with EPROM purchase.

UNITEK • P.O. Box 671 • Emporia, VA 23847

'68' MICRO JOURNAL

- ★ The only ALL 6800 Computer Magazine.
- ★ More 6800 material than all the others combined:

MAGAZINE COMPARISON

(2 years)

Monthly Averages

KB	BYTE	6800 Articles	CC	DOBB'S	TOTAL PAGES
7.8	6.4	2.7	2.2	19.1 ea. mo.	

Average cost for all four each month: \$6.53
(Based on advertised 1-year subscription price)

'68' cost per month: \$2.04

That's Right! Much. Much More

for About

1/3 the Cost!

OK, PLEASE ENTER MY SUBSCRIPTION

Bill My: Master Charge ☐ — VISA ☐

Card # _____ Exp. Date _____

For ☐ 1-Year ☐ 2 Years ☐ 3 Years

Enclosed: \$ _____

Name _____

Street _____

City _____ State _____ Zip _____

My Computer Is: _____

68 Micro Journal
5800 Cassandre Smith Rd.
Hixson, TN 37343

SUBSCRIPTION RATES

USA

1 Year \$24.50, 2 Year \$42.50, 3 Year \$64.50

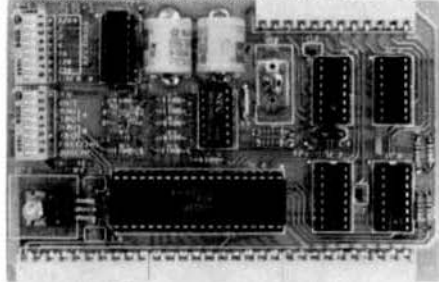
*FOREIGN SURFACE Add \$12.00 per Year to USA Price

*FOREIGN AIRMAIL Add \$36.00 per Year to USA Price

**CANADA & MEXICO Add \$5.50 per Year to USA Price
Cash (USA) or drawn on a USA Bank!!!



CALENDAR-CLOCK / TIMER / PARALLEL PORT



Calendar - Clock CLK68-1

- Reads date and time whether or not the computer is on
- All clock functions fully automatic
- On read battery (included) and standard circuit board for power
- Dot of week, month/day/year, hour/minute (12/24 hr)

Interval Timer

- Two precision oscillators, multi-tuning, etc.
- Compatible with OS-9, and Flex 1/2"
- OS-9 read timer with CLK68-1 has with timer module as 24000 Hz
- Conversion interval: interval timer module to 240 Hz

Parallel I/O Port -- Fully buffered 8 bit parallel port

- BIP interface: output or input buffer (no jumper on the board)
- Compatible with parallel port drivers to dual address at BASIC

Construction -- Fully automatic, solder mounted, 5 pin connector

Manual -- Well documented - 36 pages

Dealer & OEM discount available

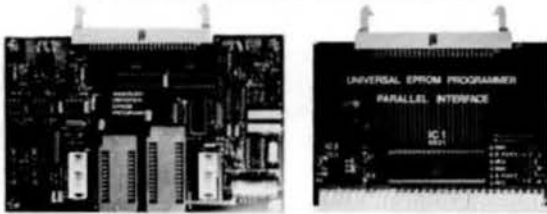
Assembled and tested	\$119.95	Kit	\$89.95
Goldplated bus conn	7.50	2 MHA option	2.50
Disk 5 or 8 in. SSB or Flex OS-9 Available, NOW			14.95

- OS-9 is a trademark of Microsoft Systems Corporation
- Flex is a trademark of Teradata Systems Corporation, Inc.

ROBERTSON ELECTRONICS Phone (505) 294-0025
1003 Warm Sands Dr. SE NM residents add 4% tax
Albuquerque, NM 87123 Add \$3 Shipping & Handling

WINDRUSH MICRO SYSTEMS

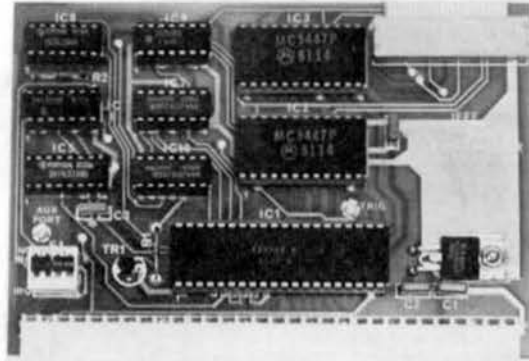
UNIVERSAL EPROM PROGRAMMER



- PROGRAMS and VERIFIES 2508, 2708, 2516, 2716, 2532, 2732A, 2564, and 2764 EPROMS. Minor hardware mods are required to program the INTEL 27128.
- Tri-volt and Single Volt 2508/2708 and 2516/2716 devices are supported.
- ZIF sockets with mode selector switches eliminate 'personality modules'.
- Twin boards with five feet of twisted pair plenum cable puts the programmer out on the bench where it belongs.
- SS-30 and EXORCISOR interfaces are available.
- Menu driven software provides the following facilities:
 - a. MOVE blocks of memory within the buffer.
 - b. READ an EPROM into the buffer.
 - c. VERIFY an EPROM against the buffer.
 - d. EXAMINE and change the contents of the buffer.
 - e. PUMP the contents of the buffer in HEX and ASCII.
 - f. FILL a selected area of the buffer with a specified character.
- Software available for all versions of SSB DOS, FLEX 2, FLEX 9 and OS-9. Assembly language source files supplied on disk.....enables customizing.
- Well documented users manual provides step-by-step adaptation and operating instructions.

AVAILABLE FROM GMIX IN THE U.S.A.

IEEE-488



- SUPPORTS ALL PRINCIPAL MODES OF THE IEEE-488 (1975/8) BUS SPECIFICATION:
 - Talker
 - Listener
 - System Controller
 - Serial Poll
 - Parallel Poll
 - Group Trigger
 - Single or Dual Primary Address
 - Secondary Address
 - Talk only...Listen only
- fully documented with a complete reprint of the KILDAUS article on the IEEE bus.
- Low level assembly language drivers suitable for 6800, 6801, 6802, 6803, 6806 and 6809 are supplied in the form of listings. These drivers have been extensively tested and are GUARANTEED to work!
- Single SS-30 board (4, 8, or 16 addresses per port), fully socketed, gold plated bus connectors, and IEEE interface cable assembly.

PL/9 EDITOR/COMPILER/DE-BUGGER

- Friendly interactive environment where you have INSTANT access to the Editor, the Compiler, and the Trace-Debugger, which, amongst other things, can single step the program a SOURCE line at a time. You also have direct access to any FLEX utility and your System Monitor.
- 250 page manual is organized as a tutorial with plenty of examples.
- Fast single pass compiler produces 8K of COMPACT and FAST 6809 machine code output per minute with no run-time overheads or license fees.
- Fully compatible with TSC text editor format disk files
- Signed and unsigned BYTES and INTEGERS, 32-bit floating point REALs.
- Vectors (single dimension arrays) and Pointers are supported.
- Mathematical expressions: (+), (-), (*), (/), modulus (%), negation (-)
- Expression evaluators: (<), (<=), (>), (>=), (<=), (>=)
- Bit operators: (AND), (OR), (EOR/XOR), (NOT), (SHIFT), (SWAP)
- Logical operators: (AND), (OR), (EOR/XOR), (NOT), (SHIFT), (SWAP)
- Control statements: IF..THEN..ELSE, IS..CASE1..CASE2..ELSE, BEGIN..END, WHILE.., REPEAT..UNTIL, REPEAT..FOR, FOR, CALL, JUMP, RETURN, BREAK, GOTO.
- Direct access to (ACCA), (ACCB), (ACCB), (CCB) and (XREG).
- FULLY supports the MC6809 SWI, SWI2, SWI3, SWI4, FIRO, FIRO2 and RESET vectors. Uniting a self-starting (from power-up) program that uses ANY, or ALL, of the MC6809 interrupts is an absolute snap!
- Procedures may be passed and may return variables. This does then functions which behave as though they were an integral part of PL/9.
- Several fully documented library function modules are supplied: I/O, BITIO, HARDIO, MATHIO, FLEXIO, SCIPACK, STRSUBS, and REALCOM.

... THIS IS THE MOST EFFICIENT COMPILER I HAVE FOUND TO DATE.

Quoted from Ron Anderson's FLEX User Notes column. Need we say more?

WE STOCK THE FOLLOWING COMPANIES PRODUCTS:
GMIX, SSB, F.H.L., MICROWARE, TSC, LUCIDATA, AND ALPORD & ASSOCIATES.

FLEX (tm) is a trademark of Technical Systems Consultants, OS-9 (tm) is a trademark of Microware Systems Corporation, ROMs (tm) and EXORCISOR (tm) are trademarks of Motorola Incorporated.

MACE/XMACE

- a co-resident EDITOR/ASSEMBLER for the 6809 written by Graham Trotti which takes most of the pain out of assembly language program development
- Friendly inter-active environment where you have INSTANT access to the Editor, the Assembler, FLEX and your System Monitor.
- MACE can also produce ASMDOC's for PL/9 with the assembly language source passed to the output file as comments.
- Includes XMACE a co-resident 6800/1/2/3/8 EDITOR/CROSS ASSEMBLER.

C

This is the FLEX version of the James McCosh 'C' compiler that is also available on UNIFLEX from SWTP and OS-9 from Microware:

- The FLEX implementation supports the full Kernighan and Ritchie 'C' specification except 'limits', 'double', and 'bit-fields'.
- Produces very efficient assembly language source output with the 'C' source optionally interleaved as comments.
- Built-in optimizer will shorten object code by about 11%
- Supports interleaved assembly language programs.
- The TSC relocating assembler/linking loader (SP09-17) is REQUIRED.

MACE	(6809 FLEX only).....	\$ 98.00
PL/9	(6809 FLEX only)...(a steel at this price!).....	\$198.00
IC1	(a 50k system and the TSC SP09-17 package is req'd)....	\$295.00
IEEE-488	with IEEE-488 cable assembly.....	\$298.00
SS-30	Universal EPROM programmer w/one version of software....	\$375.00
EXORCISOR	Universal EPROM programmer w/one version of software....	\$395.00
SOFTWARE	drivers for a 2nd, 3rd, or 4th operating system.....	\$ 25.00

ALL PRICES INCLUDE AIR MAIL POSTAGE

4n SS-50C all CMOS 256K
STATIC RAM board will be
available soon!

Write for details &
pricing.

WORSTED LABORATORIES,
NORTH WALSHAM, NORFOLK,
ENGLAND. NR28 9SA.
TEL: (0692) 405189
TLX: 97380 SHARET G

FEATURES THE
POWERFUL, THIRD
GENERATION,
MOTOROLA 6809
PROCESSOR!

THE 6809 "UNIBOARD"TM SINGLE BOARD COMPUTER KIT

PERFECT FOR COLLEGES, OEM'S, INDUSTRIAL
AND SCIENTIFIC USES!

64K RAM! DOUBLE DENSITY
FLOPPY DISK CONTROLLER!

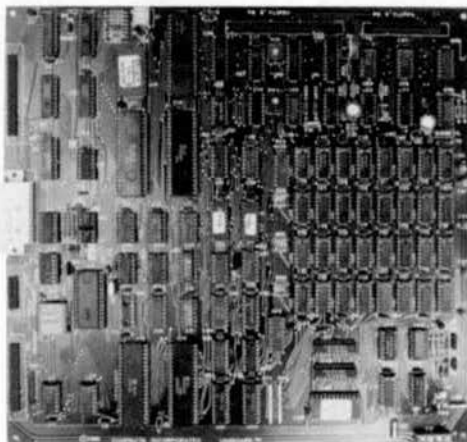
New!

BLANK PC BOARD

\$99⁹⁵

WITH PAL'S, AND
TWO EPROMS.

FOR 5-1/4 OR 8 INCH
SOURCE DISKETTE
ADD \$10.



\$399⁰⁰

COMPLETE KIT!
FULLY SOCKETED.

ALL OPTIONS ARE
STANDARD. NO
EXTRAS TO BUY!

THE COMPACTA UNIBOARDTM: Through special arrangement with COMPACTA INC., we are proud to have been selected the exclusive U.S. Mfg. of their new 6809 UNIBOARDTM COMPUTER KIT. Many software professionals feel that the 6809 features probably the most powerful instruction set available today on ANY 8 bit micro. Now, at last, all of that immense computing power is available at a truly unbelievably low price.

FEATURES:

- ★ 64K RAM using 4116 RAMS.
- ★ 6809E Motorola CPU.
- ★ Double Density Floppy Disk Controller for either 5-1/4 or 8 inch drives. Uses WD1793.
- ★ On board 80 x 24 video for a low cost console. Uses 2716 Char. Gen. Programmable Formats. Uses 6845 CRT Controller.
- ★ ASCII keyboard parallel input interface. (6522)
- ★ Serial I/O (6551) for RS232C or 20 MA loop.
- ★ Centronics compatible parallel printer Interface. (6522)
- ★ Buss expansion Interface with DMA channel. (6844)
- ★ Dual timer for real time clock application.
- ★ Powerful on board system monitor (2732). Features commands such as Go To, Alter, Fill, Move, Display, or Test Memory. Also Read and Write Sectors. Boot Normal, Unknown, and General FlexTM.

YOUR CHOICE OF POPULAR DISK OPERATING SYSTEMS:

FLEX TM from TSC	\$149
OS9 TM from Microware	\$199
Specify 5-1/4 or 8 Inch	

PC BOARD IS
DOUBLE SIDED, PLATED THRU
SOLDER MASKED, 11 x 11-1/2 IN.

ALL SALES ARE MADE SUBJECT TO THE TERMS OF OUR 90 DAY
LIMITED WARRANTY. A FREE COPY IS AVAILABLE UPON REQUEST.

Digital Research Computers
(OF TEXAS)

P.O. BOX 461565 • GARLAND, TEXAS 75046 • (214) 271-3538

TERMS: Shipments will be made approximately 3 to 6 weeks after we
receive your order. VISA, MC, cash accepted. Add \$4.00 shipping.
USA AND CANADA ONLY

64K SS-50 STATIC RAM

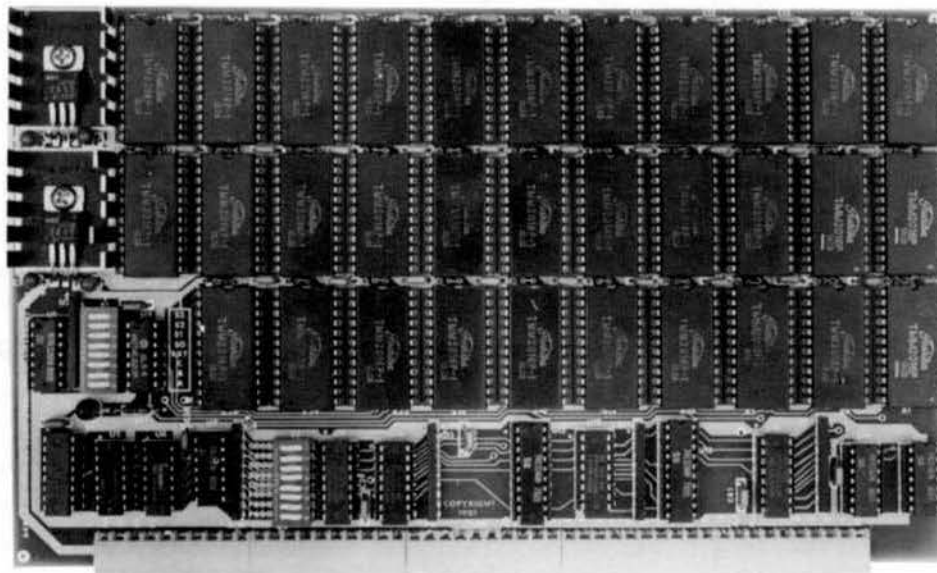
\$179⁰⁰
(48K KIT)

NEW!

NEW!

LOW
POWER!

RAM
OR
EPROM!



BLANK PC BOARD
WITH DOCUMENTATION
\$52

SUPPORT ICs + CAPS - \$18.00
FULL SOCKET SET - \$15.00

ASSEMBLED AND TESTED ADD \$50

FEATURES:

- ★ Uses new 2K x 8 (TMM 2016 or HM 6116) RAMs.
- ★ Fully supports Extended Addressing.
- ★ 64K draws only approximately 500 MA.
- ★ 200 NS RAMs are standard. (TOSHIBA makes TMM 2016s as fast as 100 NS. FOR YOUR HIGH SPEED APPLICATIONS.)
- ★ Board is configured as 3-16K blocks and 8-2K blocks (within any 64K block) for maximum flexibility.
- ★ 2716 EPROMs may be installed anywhere on Board.
- ★ Top 16K may be disabled in 2K blocks to avoid any I/O conflicts.
- ★ One Board supports both RAM and EPROM.
- ★ RAM supports 2MHZ operation at no extra charge!
- ★ Board may be partially populated in 16K increments.

56K	\$219
64K	\$249

16K STATIC RAMS?

The new 2K x 8, 24 PIN, static RAMs are the next generation of high density, high speed, low power, RAMs. Pioneered by such companies as HITACHI and TOSHIBA, and soon to be second sourced by most major U.S. manufacturers, these ultra low power parts, feature 2716 compatible pin out. Thus fully interchangeable ROM/RAM boards are at last a reality, and you get BLINDING speed and LOW power thrown in for virtually nothing.

CLOSE OUT SPECIAL
WE HAVE DROPPED OUR 32K SS-50 STATIC RAM BOARD WHICH USED 2114 LOW POWER RAMS. WE WILL SELL THE REMAINING STOCK OF BLANK PCB'S WITH DATA FOR \$17.50 EA. THESE FORMERLY SOLD FOR \$50.

Digital Research Computers
(OF TEXAS)

P.O. BOX 401565 • GARLAND, TEXAS 75040 • (214) 271-3538

TERMS: Add \$2.00 postage. We pay balance. Order under \$15 add 75c handling. No. C.O.D. We accept Visa and MasterCard. Tex. Res. add 5% Tax. Foreign orders (except Canada) add 20% P & H. Orders over \$50, add 85¢ for insurance.

OS/9™, FLEX™, UNIFLEX™, IBM PC™ Software

SUPER SLEUTH DISASSEMBLER \$99-FLEX \$100-UNIFLEX \$101-OS/9

This program processes 6 00/11/2/3/5/8/9 8502 programs, enabling the user to interactively analyze, modify, and disassemble (with labels) object code, with output to terminal, printer, and disk, and cross-reference and label definition capabilities. Object-Only Version for Color FLEX \$60, COCO OOB \$99.

Z-80/8080/5 SUPER SLEUTH DISASSEMBLER \$99-FLEX \$100-UNIFLEX \$101-OS/9

This version of SUPER SLEUTH processes Z-80/8080/5 object code on the 6800/1/9.

CROSS ASSEMBLERS each \$50 3/\$100-FLEX each \$60 5/\$120-UNIFLEX each \$55 3/\$110-OS/9

These programs and macros enable the user to process 6 00/1, 8805, Z-80, 8080/5 programs in original format.

The TSC macro assembler is required for FLEX/UNIFLEX and the RMA assembler is required for OS/9.

[14]6805 and 6502 DEBUGGING SIMULATORS each \$75-FLEX \$80-UNIFLEX \$100-OS/9

These programs enable the user to interactively analyze, modify, and debug [14]6805 and 6502 object code.

6502-TO-6809 XLATOR SYSTEM \$75-FLEX \$80-UNIFLEX \$85-OS/9

This program enables the user to translate 6502 assembler code into 6809 assembler code, noting inexact conversions.

6800-6809 & 6809 PIC XLATORS both \$50-FLEX \$60-UNIFLEX \$75-OS/9

These programs enable the user to translate 6800/1 assembler programs to 6 09 mnemonics and to convert 6 09 programs to position-independent code and data, using PC, S, U, X, and Y as base registers.

OS/9 and UNIFLEX SIMULATORS FOR FLEX each \$100-FLEX

The programs enable the user to debug OS/9 and UNIFLEX assembler programs using the TSC DEBUG and other facilities of FLEX.

DISK UTILITY PROGRAMS all \$50-FLEX

These programs enable the user to list/m modify the SIR, to edit sectors, to test entire diskettes, to linearize the free list, to back up one disk to another, etc.

FULL SCREEN FORMS DISPLAY (6809 X-BASIC) \$50-FLEX \$75-UNIFLEX \$60-IBMP

These programs enable the user to define and generate table-driven full-screen display and data-entry programs.

FULL SCREEN MAILING LIST (6809 X-BASIC) \$100-FLEX \$110-UNIFLEX \$105-IBMP

These programs enable the user to define and maintain mailing-list-oriented data bases.

FULL SCREEN INVENTORY/MRP (6809 X-BASIC) \$100-FLEX \$120-UNIFLEX \$110-IBMP

These programs enable the user to define and maintain inventories, and include hierarchical materials requirement planning.

TABULA RASA SPREADSHEET (6809 X-BASIC) \$100-FLEX \$125-UNIFLEX

These programs enable the user to generate and maintain tabular computation schemas, providing a simple user interface and sophisticated report-generation, similar to DESKTOP/PLAN (TM Desktop Computing).

TSC BASIC/XPC UTILITY PROGRAMS all \$25-FLEX \$50-UNIFLEX \$30-IBMP

These programs enable the user to resequence or cross-reference any Basic Program and generate XPC Basic sort programs.

Programs in source on disk - specify size, sides, density, type, computer, O/S.

Detailed printed manuals provided with all products.

For VISA and MASTER CARD give account, exp date, phone. US funds only - add 5% (10% foreign) for shipping.

Open Purchase Orders for D and B rated clients only. Call or write for catalog and dealer information.

*FLEX is a trademark of Technical Systems Consultants. *OS9 is a trademark of Microware.

Computer Systems Consultants, Inc.

1454 Letta Lane, Conyers, GA 30207

Telephone Number 404-483-1717/4570

LOOK
FOR
THIS
LABEL!

XDMS Data Management System
Copyright © 1983 by IBM Corp.
Version 1.1.1 Released 10-20-83

Note: Pre-8/83 shipments do not have the boldface "XDMS"

This label is your assurance that you have an ORIGINAL copy of the XDMS system. It is also your Proof of Purchase which entitles you to technical support, notification of changes and enhancements, and system upgrades at reduced rates (Version upgrades are \$25.00 plus shipping).

We register all XDMS sales in our user database (implemented under XDMS, of course). This provides us with both a mailing list, and an inventory of users. We know who purchased what version and when. By comparing dates with our modification log, we can tell you what changes effected your system, and what has occurred since you purchased your system. When applicable, we will recommend a later copy within a version and will update your disk for a small service fee of \$5.00 plus shipping. (\$10.00 if we supply a new disk).

We are sensitive to your needs. If you require a specific function - and it is GENERIC - we will investigate implementation. We are USER DRIVEN! XDMS owes much of its flexibility to user requests. In the first six months of circulation, eight enhancements were at the request of users.

XDMS is personalized to you or your business. Your name or business name appears as part of the licensing and copyright messages which appear when XDMS is invoked. If yours is missing or incorrect, you have a bogus copy - contact us with your supplier's name - We owe us money and owes you an original copy! (Note: Authorized Dealers are permitted to loan out their copies PROVIDED THAT your order has been placed with us!)

These measures are to assure you - the end user - of the highest quality and standards possible. We believe in supporting our product. We dislike getting surprise phone calls from unknown users (which has happened). We fully recommend our product and believe it's the best available. In fact, XDMS has evolved over three years to rival DBMS's available for other (non-68XX) systems which cost \$500, and up. You know who we mean. We believe the 68 family of processors is THE BEST, and will continue to provide the most flexible, easy-to-use software possible.

Should you "look for this label?" We think so. Because it stands for more than just a good piece of software. It stands for quality, for support, and above all, for a community of users who may pride themselves with excellence in their selection of both hardware and software...

WESTCHESTER Applied Business Systems
Post Office Box 187
Briarcliff Manor, N.Y. 10510

XDMS DATA MANAGEMENT SYSTEM

The XDMS (eXtended) Data Management System goes beyond the traditional functionality offered by database management systems. It is capable of the normal tabular data manipulations, calculations, sorting, searching and report generation. XDMS adds to these basic features the ability to intermix file data with text to produce custom invoices, contracts, and personalized business letters on a mass production or selective basis. XDMS defines a relational structure whereby two files may be linked together via a third. This aspect may be employed to correlate report data on a many-to-one-to-many basis. Advanced record selection permits "driving" reports and inquiries with a file derived from a secondary selection process. All this functionality is available within a single process, controlled by a single user written English-like control file.

XDMS supports sequential, hierarchical and random file structures and employs data compression to permit large files on limited disk space. The hierarchical format further reduces space required. If custom processing or file transfer are required, XDMS files may be easily translated to text format. XDMS files may be defined with up to 24 fields and record lengths up to 255 bytes. Alphanumeric, numeric, decimal, integer, scalar date, coded and hexadecimal field types are supported. Potential keys to other files are simply defined by similar field names, types and sizes. The actual link is established during processing, so that no other preparation is required.

XDMS file input and updating are accomplished via an easy to use facility which offers "editor-like" control, plus field copy, repeat function, command strings and user defined macros, all invoked by a single keystroke. Tabular and vertical screen modes accommodate most systems including those with "narrow" screen widths. The extended forms feature permits optional user defined control sequences (in assembler) which may be used to control inverse video or highlighting, underlining and other "fancy" functions on the terminal and printer.

XDMS has been designed to provide the user with maximum functionality, and yet be easy to use. No programming experience is required since all processing functions are controlled by high-level non-procedural commands. Whether your needs are for simple list-oriented applications or complex multi-file database networks, XDMS is the answer!

XDMS Data Management System (5" or 8" disk)	\$179.95
XPAK set of CMS utilities (requires XDMS)	\$79.95
XDMS Data Management System plus CMS utilities	\$249.95
XACC General Accounting System (requires XDMS)	\$249.95
Also available:	\$99.95
DM52/VM Data Management System (5" or 8" disk)	\$59.95
Util Set of 20 General Utilities (Non-DMS)	

WESTCHESTER Applied Business Systems
Post Office Box 187, Briarcliff Manor, N.Y. 10510

All software is written in macro/assembler and runs under 6809 FLEX O/S. Terms: Check, Money Order, Visa or MasterCard. Shipment first class. Add P&H \$2.50 (\$7.50 Foreign). NY Res add sales tax. Specify 5" or 8".

Sales: S. E. MEDIA, 1-800-338-6800, Consultation: 914-941-3552 (evens).

FLEX is a trademark of Technical Systems Consultants, Inc.

AMX

Real-Time Multitasking Executive

for
8080, Z80, 6809
and 8086

Gives your application a head start

Save time and money in the development of your product or system by using AMX, the software executive with proven, fault-free operation.

SIMPLE OPERATION

You divide complex control programs into a number of separate, more manageable programs, called *tasks*, each designed to do one job. This allows tasks to be written and tested separately and then combined to form a reliable, finished system.

AMX supervises the orderly execution of these tasks, assuring that the most important jobs always get done first. Tasks appear to be executing simultaneously. It's almost like having a separate CPU for each task!

HARDWARE INDEPENDENCE

AMX does not require a particular hardware configuration. *You* control your environment. *You* pick the I/O method. *You* decide the preferred interrupt service technique for your system. AMX will support you on the microprocessor of your choice.

AMX is fast, compact, and ROMable. The AMX nucleus, less than 1400 bytes in size, features multiple task priorities, intertask message passing with priority queuing, external event synchronization, and interval timing.

Support modules provide extended memory management, buffer control and resource allocation. Fast, re-entrant integer and floating point math libraries are also available.

AMX interfaces support programs written in C, PASCAL, PL/M, FORTRAN and assembler.

Access to CP/M® disk files in real time is possible using the AMX I/O Supervisor.

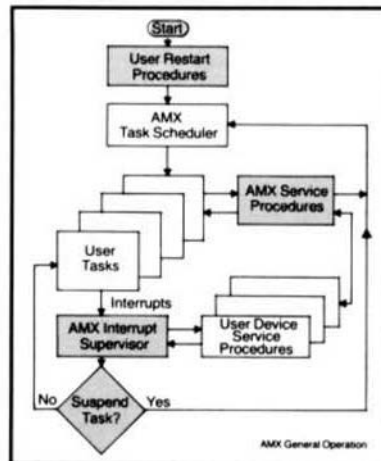
COMPLETE DOCUMENTATION

We deliver AMX source on diskette to permit AMX to be moved to the software development system of your choice. Our liberal license agreement permits binary (object) distribution without royalties.

HOW TO ORDER

A specification sheet and price list are available, free. Your check or money order for \$75 will purchase the AMX Reference Manual for immediate evaluation (specify 8080, Z80, 8086 or 6809 processor). Add \$25 for postage and handling outside USA and Canada. The standard 8080/Z80 AMX Multitasking Executive package, including source code, is ~~\$800~~. Support modules and interfaces are available separately.

AMX is the choice of professionals the world over. Make it yours, today.



AMX is a trademark of KADAK Products Ltd.
CP/M is a trademark of Digital Research Corp.
Z80 is a trademark of Zilog Corp.



KADAK Products Ltd.



206-1847 W. Broadway Ave., Vancouver, B.C., Canada V6J 1Y5/Phone: (604) 734-2796 Telex: 04-55670

6809 Word Processing System

*stylograph*TM

STYLOGRAPH 2.0
The "User Friendly" word processing system. Fewer key strokes by the operator make it easier to learn.
OS9, FLEX \$295 UnifLEX \$395
COLOR COMPUTER FLEX \$195

SPELLING CHECKER
Checks all words against an internal user-expandable dictionary of over 42,000 words.
OS9, FLEX \$145 UnifLEX \$195

MAIL MERGE
Inserts names and addresses into form letters and mailing lists. Appends files at print out time. Handles files longer than memory.
OS9, FLEX \$125 UnifLEX \$175

Inquire about our other software
• Business Programs - G/L, A/R, A/P
• Data Base Management System
• Assemblers
Also, Daisy Wheel Printers \$599.

Great Plains Computer Company Inc.
P.O. Box 916
Idaho Falls, Idaho 83401
(208) 529-3210

Flex and Uniflex are trademarks of Technical Systems Consultants, Inc.
OS9 is a trademark of Microware.

UNEQUAL[®] in the 6809 ARENA

AM-100[™] ADVANCED WINCHESTER SubSystems for FLEX[™] and OS-9[™]

COMPLETE[™] AM-100 starting as low as \$1899.-

• Intelligent microprocessor based controller • extensive error diagnostics • standard SASI interface • host interface assures efficient data transfer via hardware implemented handshaking • powerful software drivers - complete floppy/winchester integration, logical division of winchester up to 4 logical drives • quick & easy installation; no modification required to user's hardware or software • GET RID OF ANNOYING DISK ERRORS • NO MORE SLOW FLOPPY OPERATIONS •

***UNEQUAL QUALITY • UNEQUAL PRICES • UNEQUAL SUPPORT
TO OUR USERS AND DEALERS • CALL AND FIND OUT**

complete[™] subsystems

5MB FIXED -	STD. 5" -	\$1899.-
10MB FIXED -	STD. 5" -	\$2099.-
5MB FIXED -	MINI 5" -	\$2315.-
5MB REMOVABLE -	MINI 5" -	\$2395.-

**** COMPLETE SUBSYSTEM = Drive(s) + Controller + Power Supply + Enclosure + All Cables + Host Interface + Software Drivers NO HIDDEN EXTRAS !!!**

options

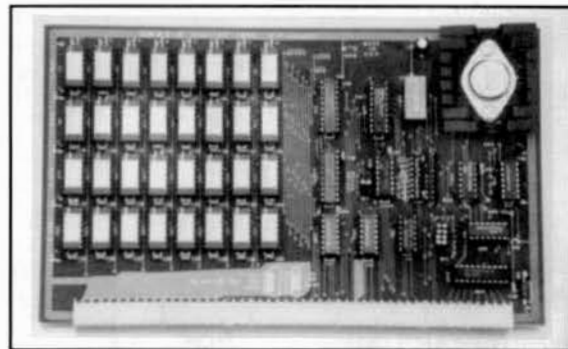
SS-50 HOST INTERFACE incl. SOFTWARE DRIVERS	\$325.-
SET OF WINCHESTER SOFTWARE UTILITIES	\$40.-
HOW TO WINCHESTERIZE - A COMPLETE MANUAL	\$15.-
shipping & handling 2.00 (US, Canada) - 4.00 (overseas)	

TERMS: • VISA • MONEY ORDER •



interfacing technologies corp.
P.O. BOX 578 Snowdon (514) 737-8787
4890 Bourret Ave. Montreal Quebec, H3X 3T7

Processors: 1. I.B. Interfacing Technologies Corp. 2. Technical Systems Consultants Inc. 3. Microware Inc.



QMM2 \$795 **256K DYNAMIC MEMORY**

Works in SS50-C 6809 systems by SWTPC, SSB, GIMIX, and others. Runs at 1 or 2 Mhz with on board refresh. Comes assembled, tested, burned in and warranted for 1 year.

Professional quality board with socketed components and gold connectors.

Also available with 64K, 128K, or 192K.

Delivery: Stock—2 weeks.

Terms: Prepaid, C.O.D., VISA, MASTERCARD.

D.P. Johnson (503) 244-8152
7855 S.W. Cedarcrest St., Portland, OR 97223

SUPER CPU

... only from
LSI



Yes, it's SUPER-CPU, advanced, new 32 bit processor with power and ability far beyond those of normal CPU's. SUPER-CPU, who can change the course of mighty SS-50 processing, break 68XX benchmarks with its bare MC68008 CPU. And who, disguised as LSI's S68K/08-CPU SS-50c compatible card, fights a never ending battle for speed, power and the SS-50c user.

We're mighty proud of our new processor card. We're giving you the ability to go 68000 without major changes to your system. Our new CPU gives you these advanced features:

- Dynamic partitioning memory management unit with bound check register.
- On-board timer for multi-user/multi-tasking applications.
- On-board boot-strap EPROM and Monitor EPROM space.
- Vectored priority interrupt generator.
- On-board wait state generator.
- User selectable bus options that includes a new higher bandwidth bus mode.
- And many more...

S68K/08-CPU..... **\$449.95**

CP/M-68K.....\$350.00

(contact factory for disk controllers supported)

CP/M-68K is a registered trademark of Digital Research, Inc



N.Y. residents add sales tax

LSI Enterprises Ltd.

PO Box 1227

Woodhaven, NY 11421

(212) 423-5596

68 MICRO JOURNAL PROGRAMS on DISK

- Disk #1: FILESORT, MINICAT, MINICOPY, MINIFMS, **LIFETIME, **POETRY, **FOODLIST, **DIET.
 Disk #2: DISKEDIT w/ Inst. & fixes, PRIME, **PRMOD, **SNOOPY, **FOOTBALL, **EXPANW, **LIFETIME.
 Disk #3: CBUG09, SEC1, SEC2, FIND, TABLE2, INTXT, DISK-EXP, **DISKSAVE.
 Disk #4: MAILING PROGRAM, *FINDDAT, *CHANGE, *TESTDISK.
 Disk #5: *DISKFIX 1, *DISKFIX 2, **LETTER, **LOVESIGN, **BLACKJAK, **BOWLING.
 Disk #6: **PURCHASE ORDER, INDEX (Disk file Indx).
 Disk #7: Linking Loader & RLOAD, Harkness
 Disk #8: CRTSET, Lonpher (May '82)
 Disk #9: DATECOPY, DISKFIX9 (Aug '82)

NOTE: All are as published or received by 68 Micro Journal, some have fixes and patches.

This is a reader service only! No warranty is offered or implied, they are as received and are for reader convenience ONLY. Also 6800 and 6809 programs are mixed, as each is fairly simple (mostly) to convert to the other.

PRICE: 8" Disk \$19.95 - 5" Disk \$17.95

68 MICRO JOURNAL
 POB 794
 Hixson, TN 37343
 615-842-4600

* Indicates 6800, ** Indicates BASIC SWTPC or TSC - 6809 no indicator.

MASTER CARD - VISA accepted - Foreign add sufficient postage surface or airt!

ANDERSON COMPUTER CONSULTANTS & Associates

Ron Anderson, respected author and columnist for 68 MICRO JOURNAL announces the **Anderson Computer Consultants & Associates**, a consulting firm dealing primarily in 68XX(X) software design. Our wide experience in designing 6809 based control systems for machine tools is now available on a consultation basis.

Our experience includes programming machine control functions, signal analysis, multi-axis servo control (CNC) and general software design and development. We have extensive experience in instrumentation and analysis of specialized software. We support all popular languages pertaining to the 6809 and other 68XX(X) processors.

If you are a manufacturer of a control or measuring package that you believe could benefit from efficient software, write or call Ron Anderson. The fact that any calculation you can do with pencil and paper, can be done much better with a microcomputer. We will be happy to review your problem and offer a modern, state-of-the-art microcomputer solution. We can do the entire job or work with your software or hardware engineers.

Anderson Computer Consultants & Associates
 3540 Shurbridge Court
 Ann Arbor, MI 48105

MACROPLEX Software

175 Fifth Avenue, Suite 3011, New York, NY 10010

Presenting... **THE CONDUIT** A new File Type for OS9

THE CONDUIT lets programs exchange data without using external files. Direct program-to-program data transfers on any path use OS9 I/O calls. No need to change device-independent programs. Two-way transfers are permitted and device-dependent code can be accommodated. Programs that formerly ran sequentially can now run concurrently to the limit of memory.

Utilities supplied with **THE CONDUIT** give immediate results when linked to your current programs via **CONDUIT** paths:

- Fast disk store-down speeds up assemblies and other I/O bound procedures
- "in-flight" terminal I/O redirection (e.g., set traps in DEBUG from a disk file)
- Basic09 source inclusions, enhanced listing
- "Null files" drain output, signal EOF on input

Free brochure available. Write, or call (212) 686-3036.

Manual alone (50 pages)	\$ 8
CONDUIT file system + utilities	\$70
Both together	\$75
Postage outside North America add	\$ 5

Specify 5" or 8" disk

Version for **OS9 Level Two** available soon

OS9™ Basic 09™ Microware Systems Corp. & Motorola, Inc.

NOW "C" FOR 6800

announcing

A SMALL -C COMPILER FOR 6800 FLEX SYSTEMS

ONLY \$29.95 INCLUDING FULL SOURCE CODE

ALSO A CPM-FLEX FILE TRANSFER UTILITY. COPIES FILES IN BOTH DIRECTIONS (DMAF1 CONTROLLER)

\$39.95 INCLUDING SOURCE CODE (6800)

both available on 8 inch FLEX diskette

FOR ALL YOU 6800 HOLDOUTS (ALSO 6809 USERS)
WE HAVE A SUPER 68008 BOARD COMING!!
(WAIT TILL YOU SEE THE PRICE!!!)

STS ENTERPRISES

5489 ARLENE WAY, LIVERMORE, CA 94550
(415) 455-6358

• California residents add 6% tax

SOFTWARE FOR THE **HARDCORE**

** FORTH PROGRAMMING TOOLS from the 68XX&X **
** FORTH specialists — get the best!! **

NOW AVAILABLE — A variety of rom and disk FORTH systems to run on and/or do TARGET COMPILE for

6800, 6301/6801, 6809, 68000, 8080, Z80

Write or call for information on a special system to fit your requirement.

Standard systems available for these hardware—

EPSON HX-20 rom system and target compiler
6809 rom systems for SS-50, EXORCISER, STD, ETC.
COLOR COMPUTER
6800/6809 FLEX or EXORCISER disk systems.
68000 rom based systems
68000 CP/M-68K disk systems, MODEL 11/12/16

tFORTH is a refined version of FORTH Interest Group standard FORTH, faster than FIG-FORTH. FORTH is both a compiler and an interpreter. It executes orders of magnitudes faster than interpretive BASIC. MORE IMPORTANT, CODE DEVELOPMENT AND TESTING is much, much faster than compiled languages such as PASCAL and C. If Software DEVELOPMENT COSTS are an important concern for you, you need FORTH!

firmFORTHSM is for the programmer who needs to squeeze the most into roms. It is a professional programmer's tool for compact rommable code for controller applications.

* tFORTH and firmFORTH are trademarks of Talbot Microsystems
* FLEX is a trademark of Technical Systems Consultants, Inc.
* CP/M-68K is trademark of Digital Research, Inc.

tFORTHSM from TALBOT MICROSYSTEMS NEW SYSTEMS FOR 6301/6801, 6809, and 68000

---> tFORTH SYSTEMS <---

For all FLEX systems: GIMIX, SWTP, SSB, or EXORCISOR Specify 5 or 8 inch diskette, hardware type, and 6800 or 6809.

- ** tFORTH — extended fig FORTH (1 disk) \$100 (\$15)
with fig line editor.
- ** tFORTH + — more! (3 5" or 2 8" disks) \$250 (\$25)
adds screen editor, assembler, extended data types, utilities, games, and debugging aids.
- ** TRS-80 COLORFORTH — available from The Micro Works
- ** firm FORTH — 6809 only. \$350 (\$10)
For target compilations to rommable code.
Automatically deletes unused code. Includes HOST system source and target nucleus source. No royalty on targets. Requires but does not include tFORTH +.
- ** FORTH PROGRAMMING AIDS — elaborate decompiler \$150
- ** tFORTH for HX-20, in 16K roms for expansion unit or replace BASIC \$170
- ** tFORTH/68K for CP/M-68K 8" disk system \$290
Makes Model 16 a super software development system.
- ** Nautilus Systems Cross Compiler
— Requires: tFORTH + HOST + at least one TARGET:
— HOST system code (6809 or 68000) \$200
— TARGET source code: 6800-\$200, 6301/6801-\$200
same plus HX-20 extensions— \$300
6809—\$300, 8080/Z80—\$200, 68000—\$350

Manuals available separately — price in ().
Add \$6/system for shipping, \$15 for foreign air.

TALBOT MICROSYSTEMS 1927 Curtis Ave., Redondo Beach, CA 90278 (213) 376 9941

SS-50 SOFTWARE

FROM THE WORLD'S LARGEST SUPPLIER OF SOFTWARE
FOR FLEX AND OS-9

A/BASIC COMPILER - generates pure, fast efficient 6809 machine code from easy to write BASIC source programs. For 6809 OS-9 or FLEX - \$150.00

DYNASTAR - Screen Editor - A powerful menu-driven screen editor specially suited to the needs of program preparation and on-line processing. For 6809 OS-9 or FLEX - \$100.00

DYNAPROB - Test Programmer - Provides all the debugging features such as pagination, headers and footers with page numbers, single steps, double steps, hold for double steps and more. Purchase with DYNASTAR for complete word processing station. For 6809 OS-9 or FLEX - \$140.00

DYNASOFT PASCAL - A fast integer only P-Code compiler ideal for 6809 applications. Powerful enough for the DYNASTAR word processor. Written for 6809 OS-9 - \$40.00 or FLEX - \$50.00 (Source code is additional \$25.00)

DTRA "C" - A very fast mode "C" from the people who brought you Dynastar. For 6809 OS-9 - \$100.00

ELCP - A data retrieval utility designed to save you time digging through manuals looking for the exact computer language commands and statements. Written in assembler for 6809 FLEX - \$20.00

ASSEMBLE - Contains a set of memory resident commands that are added in the design of user oriented applications programs. Written in assembler for 6809 FLEX - \$10.00

W/CONVAND - A utility for converting disk resident commands to memory resident commands \$20.00 (Free with purchase of Assembler)

TOIR - Provides a way for FLEX users to test advantage of a large memory chip. Requires user want to test extended memory as an additional step driver. Written for 6809 FLEX - \$100.00

ASSEMBLER - A tool for all artists and technicians. It reads your text and code you send and it will write it. In assembler for 6809 FLEX - \$20.00

CRASH - Will cross assemble source code into object code. The source or binary available for the following: 6800, 6801, 6802, 6803, 6805, 6806, and 6808. OS-9 version includes 6809 binary. (Use 6809 system to develop software for 6809 Micro Color Computer) OS-9 and FLEX \$100.00. CPU's additional.

OS-9 - A fast and reliable memory management with ability to online memory and additional protection, conditional assembly directives and ability to change value of a label or symbol. Creates OS-9 binary files in 16K and vice versa. For OS-9 or FLEX - \$100.00

SW/ASM - ED has all the features of TECO BETA2 with the addition of screen type editing, macro capability, and a main package. ASM is comparable with TECO ASSEMBLER, but with more powerful macros and modifiable OS-9 TECO. For 6809 FLEX - \$60.00

DYNASPELL - The most versatile 6809 spelling checker available. Fast and easy to use. In assembler for 6809 OS-9 or FLEX - \$100.00

PL-2 - A convenient EDITOR COMPILER/DEBUGGER option to Graham Trett. A single pass compiler that performs position independent machine code output. For 6809 FLEX - \$100.00 (Exclusive US distribution)

SACS - A convenient 6809 ASSEMBLER written by Graham Trett. Includes SACS, a 6809 assembler 6809/73 CTR/CTR COMP ASSEMBLER. For 6809 FLEX - \$60.00

FLEX EXTENDED USE UTILITIES - A set of 12 utilities that add the final touch to your editor for FLEX. Written in assembler for 6809 FLEX - \$40.00

TOOLSET #1 - Add 1077 capabilities to TECO SACS, also with DISCOMPIL and cross reference. For 6809 FLEX - \$40.00

TOOLSET #2 - A package of utilities and programs developed to improve the capabilities of the FLEX operating system. Written in assembler for 6809 FLEX - \$40.00

ARCADE 50

POWERFUL COLOR GRAPHICS
Uses the new TMS9918A Video Display processor. High resolution 256 x 192 pixel display with 15 colors. 16K Bytes of onboard RAM does not reduce user memory. 32 graphic images can be individually moved with simple X-Y commands for smooth animation. External Video input allows subtitling. NTSC composite video output.

SOUND EFFECTS AND MUSIC
• Thr AY3-8910 Programmable Sound Generators
• Nine simultaneous voices
• Thr independent noise sources
• Onboard stereo amplifier drives two 8 ohm speakers

ADDITIONAL I/O CAPABILITIES
• Eight analog inputs with 8 bit resolution
• Supports four jo sticks with pushbutton switches
• Eight bit parallel I/O port
• Entire unit maps into 256 bytes of memory

FBASIC

TERMINUS DESIGN INC. in conjunction with Microware Systems Corporation, is proud to announce FBASIC an enhancement of Microware's 6800/ BASIC. Their fast compiled BASIC has been adapted for 6809 users with added video and sound features for ARCADE 50 users. FBASIC is a true compiler that produces optimized machine language modules which are ROMable and require no Run-Time package. FBASIC requires less memory overhead and runs hundreds of times faster than BASIC interpreters. It supports standard BASIC instruction including String functions. Disk I/O and fast integer arithmetic with multiple-precision capability. Graphics verbs and functions fully support the Arcade 50.

AR ADE 50 assembled and tested	\$325.00
Video and Audio connectors	15.00
4 Joyalick connector set	15.00
2 Radio Shack Jo sticks	24.00
Gold Molex connectors	12.00
A/BASIC for 6809	110.00
FBASIC for 6809	110.00
FBASIC (with APC DE 50)	75.00
ARCADE 50 RGB	375.00
LABVIDEO (Motorola EXORbus)	375.00
NEW MV09 6809 Processor Board	225.00
256K Dyn mic Memory Board	795.00
256K Dynamic Memory Board (w/64MB)	395.00
64K Dynamic Memory Board	295.00

TERMINUS DESIGN INC
16 SCARBROUGH ROAD
ELLENWOOD, CA 90049
(404) 474-4866

TERMINUS DESIGN INC. 1983

FHI FRANK HOGG LABORATORY
THE REGENCY TOWER • SUITE 215 • 770 JAMES ST. • SYRACUSE, NY 13203
PHONE (315) 474-7856 • TELEX 646740

Cut Your Design Time In Half!

Our TSL 6809/6802 MPU Emulator Allows You To Do Just That!

Tired of troubleshooting 6809/6802 based systems with an oscilloscope alone? Is the continual EPROM burning during software development slowing you down? If your answer to either question is "yes," TSL's 6809/6802 MPU Emulator could be the time-saver you have been waiting for. Our Emulator requires no hardware modifications to your SWTPC computer.

TSL's 6809/6802 MPU Emulator for Southwest Technical Computers...

- Examines external hardware addresses from the keyboard of the SWTPC computer (using the TSL's EM-BUG monitor).
- Executes in Southwest RAM development software which addresses external hardware.
- Sets hardware breakpoint for memory location: read attempt, write attempt, and access attempt.
- Sets up to 10 software breakpoints (SWI type).

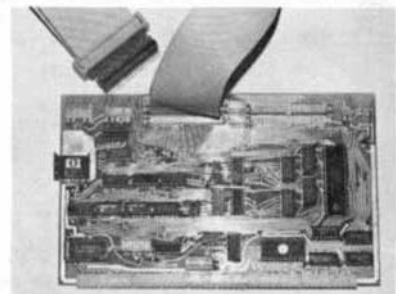
System Requirements:
SWTPC computer with 20-bit addressing
Fully decoded memory boards
FLEX™ operating system

Emulator package contains:
One SS-50 Emulator board
EM-BUG monitor (in ROM)
EM-SWTPC configuration program on disk
All interconnecting cables
User's manual

At \$695.00 TSL's EMULATOR could pay for itself with your first program. Call 314/343-9140 and ask to speak with one of our design engineers or write to: Teknar of St. Louis, 306 Axminster Drive, Fenton, MO 63026.

TERMS: VISA, MasterCard, Check or money order. Mo. residents add \$39.00 tax. Foreign countries pay in U.S. dollars. Normal Delivery: Stock to 2 weeks. Shipped U.P.S. in continental U.S.A. Best way to foreign countries.

Trademark of Technical Systems Consultants



TEKNAR, INC. OF ST. LOUIS



ACORN
COMPUTER SYSTEMS 88-50C

MODULES - BARE CARDS - KITS - ASSEMBLED & TESTED

Stackable Modules	KIT	A&T
20 AMP POWER SUPPLY w/fan w/Disk protect relay	350.00	400.00
DISK CABINET w/rege. & cables less DRIVES	200.00	250.00
MOTHER BOARD, 8 88-50c, 8 88-30c NMI button	225.00	325.00

Item	Bare	KIT	A&T
ITS - INTERRUPT TIMER 1, 10, 100 p r sec.	19.95	29.95	39.95
PB4 - INTELLIGENT PORT BUFFER Single board comput.	39.95	114.95	139.95
DP1A - Dual PIA parallel port 4 buffered I/Os	24.95	89.95	89.95
KADR - Extended Addressing BAUD gen. PIA port	29.95	89.95	89.95
MB8 - MOTHER BOARD 88-50c w/BA D gen.	84.95	149.95	199.95
P168 - 168K PROM DISK 21, 2764 EPROMs	39.95	79.95	10.95
FD88 - Firmware development 2, 8K blocks	39.95	84.95	114.95
XMPR - 2764 PROM burner ad pt. for 2716 BURNER		19.95	-----
CERRY Keyboard w/Cabinet 96 key cap citive		249.95	-----
TAXAR 12", 18 mhz MONITOR GREEN AMBER		-----	149.95 139.95
4 MODULE CABINET - unfinished POWER SUPPLY w/disk protect		150.00 250.00	-----

Color Computer

MONOLINK - 20 Mhz Monochrome video driver	15.00	20.00
CC30 PORT BUS w/power supply 5 88-30, 2 Cart	189.95	199.95
POWER BOX 6 switched outlets transient suppression	29.95	39.95
RS-232 3-switched ports for above	ADD +20.00	+25.00



Write for FREE Catalog

ADD \$3.00 S&H PER ORDER
WIS. ADD 5% SALES TAX



11931 W. Bluemound Road
MILWAUKEE, WIS. 53226
(414) 257-0300

'68' MICRO JOURNAL ADVERTISERS INDEX

'68' MICRO JOURNAL	60,67
AAA CHICAGO COMPUTER CENTER	36,37
ACKERMAN DIGITAL SYSTEMS INC.	56
ACORN COMPUTER SYSTEMS	70
ADVANCED DIGITAL TECHNOLOGY	6
ALFORD & ASSOCIATES	57
ANDERSON COMPUTER CONSULTANTS	67
CLEARBROOK SOFTWARE GROUP INC.	71
COLOR MICRO JOURNAL	9
COMPILER EVALUATION SERVICES	49
COMPUTER EXCELLENCE	56
COMPUTER PUBLISHING INC.	5
COMPUTER SYSTEMS CENTER	49,59
COMPUTER SYSTEMS CONSULTANTS, INC.	64
D.P. JOHNSON	66
DATA-COMP	18C
DIGITAL RESEARCH COMPUTERS	62,63
FISHER SCIENTIFIC	1FC
FRANK HOGG LABORATORY, INC.	69
GIMIX, INC.	3,72
GREAT PLAINS COMPUTER CO.	66
HAZELWOOD COMPUTER SYSTEMS	08C
INTERFACING TECHNOLOGIES, CORP.	66
INTROL CORP.	48
JOTO ASSOCIATES	71
KADAK PRODUCTS LTD.	65
LSI ENTERPRISES LTD.	67
MACROPLEX SOFTWARE	68
MICROWARE SYSTEMS CORP.	1,4
PERIPHERAL TECHNOLOGY	58
ROBERTSON ELECTRONICS	60
SMOKE SIGNAL BROADCASTING	7
SOUTH EAST MEDIA	50,51,52,53,54,55
SPECIALTY ELECTRONICS, INC.	58
STAR-KITS	58
STS ENTERPRISES	68
TALBOT MICROSYSTEMS	68
TEKNAR, INC.	69
TERMINUS DESIGN, INC.	69
THOMAS INSTRUMENTATION	71
UNITEK	60
UNIVERSAL DATA RESEARCH, INC.	8
WESTCHESTER APPLIED BUSINESS SYSTEMS	64
WINORUSH MICRO SYSTEMS LIMITED	61

This index is provided as a reader service. The publisher does not assume any liability for omissions or errors.

THOMAS INSTRUMENTATION

HARDWARE

CPU-5
S-R/R w/o memory chips
S-R/R with 48K
SP-1
6802 SUPER CPU
VIDEO RAM
PARALLEL I/O
SS-50/50C EXTENDER
SS-30 EXTENDER
SS-50 WIRE WRAP
SS-30 WIRE WRAP
SS-30 BACKPLANE 8POS.
SS-50 BACKPLANE 4,8,12,
& 16 @ \$5.00/SLOT

ASSEM & TEST

\$299.00
\$120.00
\$399.00
\$195.00
\$235.00
\$195.00
\$139.00
\$ 35.00
\$ 25.00
NA
NA
NA
NA

BARE CARD

\$49.00
\$49.00
\$49.00
\$49.00
\$59.00
\$49.00
\$49.00
NA
NA
\$39.00
\$20.00
\$39.00
NA

SOFTWARE

VDISK
Use extended memory
as a fast disk drive
6809 source & obj \$149.00
6809 object \$ 99.00

OUTSIDE MODEM PROGRAM
includes source
UniFLEX™ \$100.00
FLEX™ 6800 or 6809 \$ 50.00

CROSS ASSEMBLER
For 6800, 6801, 6805
Runs on 6809 FLEX™
\$150.00

COMPONENTS

GOLD MALE MOLEX \$1.60 GOLD FEMALE MOLEX \$1.60
TIN MALE MOLEX \$.40 TIN FEMALE MOLEX \$.50
146805E2P \$20.00

THOMAS INSTRUMENTATION
168 EIGHTH STREET
AVALON, N.J. 08202
(609) 967-4280

MASTERCARD, VISA, AND C.O.D. ACCEPTED. PLEASE ALLOW 3 WEEKS
FOR CHECKS TO CLEAR. CONT. USA ADD \$3.00 SHIPPING. CANADA
\$6.00. FOREIGN \$12.00. NJ RESIDENTS ADD 6% SALES TAX
FLEX and UniFLEX ARE TRADEMARKS OF TECHNICAL SYSTEMS CONSULTANTS



Clearbrook Software Group Inc.
08-9 SOFTWARE

Disk Utilities Package (CSG-DUP) \$49

Four utilities designed to save you time (and typing). **DDir**—display a directory listing of one or more directories or of an entire disk. **DCopy**—copy files in one or more directories or an entire disk with or without prompting. **DDel**—Delete files in one or more directories or an entire disk. **DAttr**—re-attribute files or change owner I.D. for files in one or more directories or an entire disk. All utilities allow wild card file specification (partial filename, attribute and own I.D.) and have the ability to use AND, OR and NOT logic for file specification relations.

DEdit \$39 (CSG-DEDT)—A screen oriented disk editor. DEdit allows the user to view and modify any readable sector on an OS-9 format disk. A terminal with clear screen and cursor addressing functions is required. Data is displayed in hexadecimal, ASCII and binary formats. Modifications to a sector are made by moving the cursor to the byte(s) to be changed, typing the correct data and issuing an Update command. (Written in machine language.)

BT9 \$59 (CSG-BT9)—A binary tree filing program. BT9 makes it easy to write BASIC09 programs in which files must be maintained in sorted order. Three types of files are supported: Index File, Indexed Data File and Non-indexed Data File. BT9 also allows for maintaining several indexes for one file. All sorting is performed on disk leaving more memory free for other programs. BT9 is written in machine language for fast operation (typical search time under two seconds for a file of 2000 records).

SPECIAL OFFER—Order 2 CSG products and deduct 10%. Order all 3 products and deduct 15%.

Send Certified Check or Money Order to: Clearbrook Software Group, Box 8000, Suite 499, Sunas, WA 98295. In Canada write: Box 8000, Suite 499, Abbotsford, B.C. V2S 1K3

Call (604) 853-9118 for information.
Specify 8 inch or 5 inch format.

FIRST CLASS

INTELLICOM INTELLigent COMmunications Program

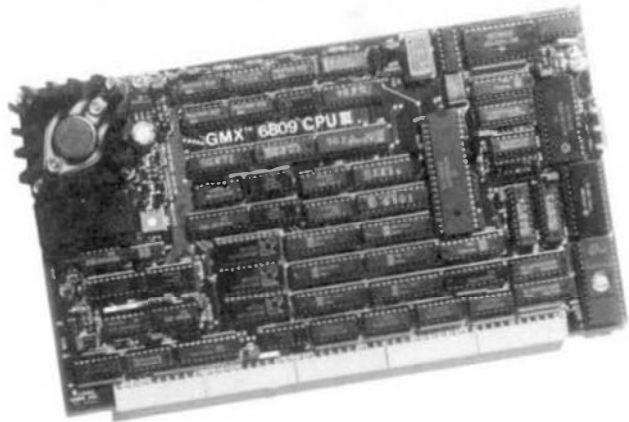
INTELLICOM provides you with the capability of Intelligent computer to computer communications from both a terminal emulation and file transfer standpoint. INTELLICOM supports several file transfer protocols that facilitate the transfer of both binary and ASCII data. Since INTELLICOM is menu drive, it is a breeze to use and understand. With INTELLICOM you will be able to communicate with the various data and timesharing services such as The Source and CompuServe. Additionally, since INTELLICOM supports the protocol used by virtually all remote CP/M systems around the world, all users can immediately begin to take advantage of the wealth of public domain software available on these systems. Current, or potential, users of CompuServe can transmit and receive both binary and ASCII data with full error detection and recovery. The checksum protocols allow for the verifications of data blocks transferred (assuming appropriate support on the host end). This feature will be of great value in those applications where data integrity is paramount. INTELLICOM's documentation includes a detailed description of all protocols used along with machine readable examples of host pseudo code. A telephone directory and dialing facility compatible with Hayes Smartmodems is supported.

*FLEX is a trademark of Technical Systems Consultants
**OS9 is a trademark of Microware

Price: OS9 or FLEX Version \$99.95, add \$5.00 COD
JOTO ASSOCIATES, 104 Ledge Dr., Southington, CT 06488

203-621-0070

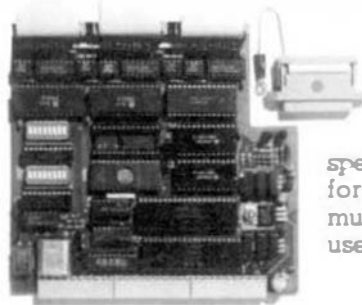
GIMIX STATE OF THE ART 6809 SYSTEMS FOR THE SERIOUS USER.



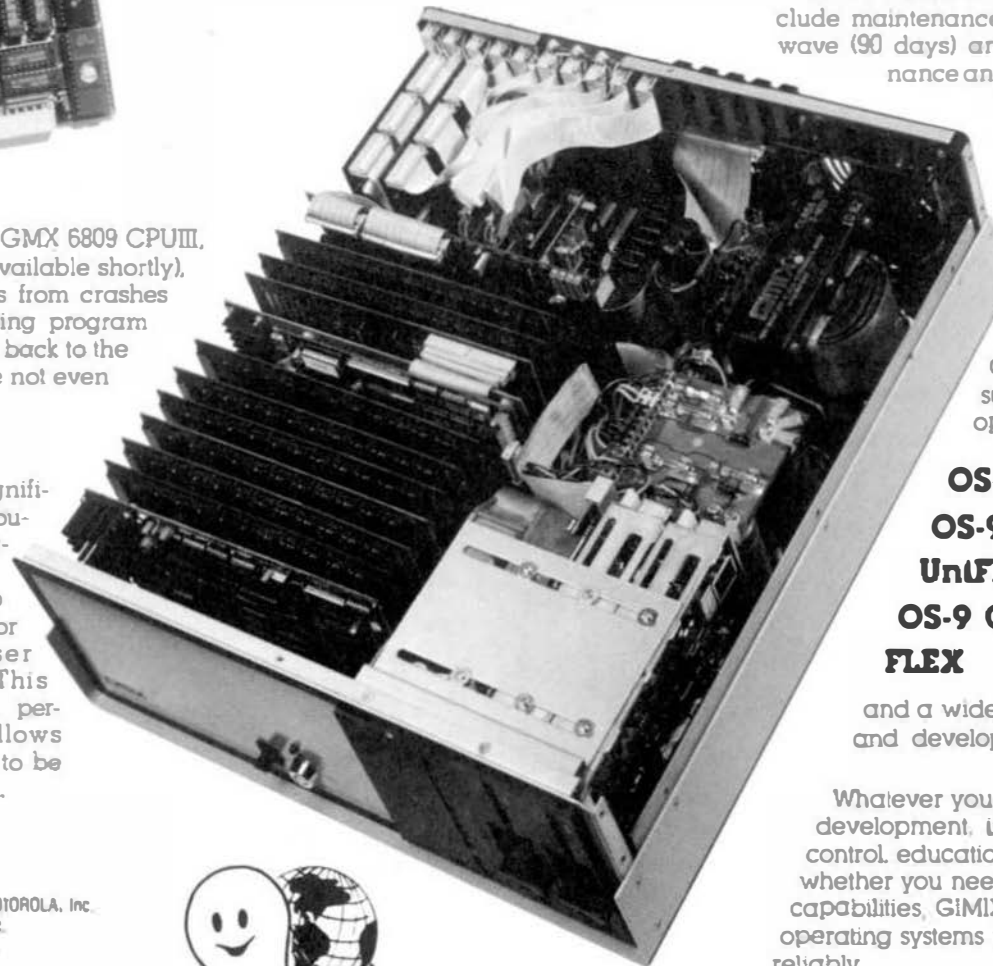
GIMIX has Floppy Disk Drive systems or can offer you 19MB or high performance 47MB Winchester Drive systems.

For the ultimate in performance, the Unique GMX 6809 CPU III, using either OS-9-GMX III or UniFLEX GMX III (available shortly), gives protection to the system and other users from crashes caused by defective user programs. e.g. During program development, a programmer who crashes goes back to the shell or the debugger, while the other users are not even aware anything occurred.

The intelligent serial I/O processor boards significantly reduce system overhead by handling routine I/O functions, thereby freeing up the host CPU for running user programs. This speeds up system performance and allows multiple terminals to be used at 19.2K baud.



BASIC-09 and OS-9 are trademarks of Microware Systems Corp. and MOTOROLA, Inc. FLEX and UniFLEX are trademarks of Technical Systems Consultants, Inc. GIMIX, GHOST, GMX, CLASSY CHASSIS, are trademarks of GIMIX, Inc.



GIMIX 6809 systems support five predominant operating systems:

**OS-9 GMX III,
OS-9 GMX II,
UniFLEX,
OS-9 GMX I,
FLEX**

and a wide variety of languages and development software.

Whatever your application: software development, instrumentation, process control, educational, scientific or business; whether you need single or multi-user capabilities, GIMIX has hardware and the operating systems to get the job done reliably.

Please phone or write if you need further information.



GIMIX inc.

1337 WEST 37th PLACE • CHICAGO, ILLINOIS 60609 • (312) 927-5510 • TWX 910-221-4055

© 1983 GIMIX Inc.

The Original FLEX™ for Color Computers

- Upgrade to 64K
- RS to FLEX, FLEX to RS file transfer ability
- Create your own character set
- Automatic recognition of single or double density and single or doubled sided
- All features available for either single or multiple drive systems
- Settable Disk Drive Seek Rates
- Faster High Resolution Video Display with 5 different formats
- Save RS Basic from RAM to Disk
- Move RS Basic to RAM
- Load and save function on FLEX disk
- 24 Support Commands 12 with Source Text
- External Terminal Program

DATA-COMP has everything you need to make your TRS-80C Color Computer WORK for YOU: from Parts and Pieces to Full Ready To Use Systems. DATA-COMP designs, sells, services and SUPPORTS Computer SYSTEMS, not just Software. CALL DATA-COMP TODAY to make your Computer WORK FOR YOU!

System Requirements

FLEX9 Special General Version + Editor & Assembler (which normally sell for \$40.00 ea) \$150.00

P-MATE(RS) FLEX9 Conversion Rout. for the RS Disk Controller \$49.95
when purchased with Special General FLEX9 Sys. \$59.95

Color Computer with 64K RAM and EXT-BASIC \$399.95

SPECIAL SYSTEM PACKAGES

64K Radio Shack COLOR COMPUTER, Radio Shack COLOR DISK CONTROLLER, a Disk Drive System, Special General Version of FLEX9, P-MATE(RS), and a Box of 10 Double Density Diskettes, a COMPLETE ready to run SYSTEM on your Color TV Set.

\$999.95

64K Radio Shack COLOR COMPUTER, Radio Shack COLOR DISK CONTROLLER, a Disk Drive System, and a Box of 10 Double Density Diskettes, a COMPLETE ready to run SYSTEM on your Color TV Set.

\$799.95

PARTS AND PIECES

Radio Shack Disk Controller \$169.95
1 Single Sided Double Density Disk Drive Random \$229.95
1 Double Sided Double Density Disk Drive Queue \$319.95
1 Queue Thinline Double Sided, Double Density \$249.95
Single Drive Cabinet with Power Supply \$79.95
Double Drive Cabinet with Power Supply \$109.95
Single Drive Disk Cable for RS Controller \$19.95

Double Drive Disk Cable for RS Controller \$24.95
Macro Tech Prods Inc. LOWER CASE ROM Adapter \$74.95
Radio Shack BASIC Version 1.1 ROM or 1.2 ROM \$34.95
Radio Shack EXTENDED BASIC ROM \$74.95
Radio Shack Disk Basic ROM 1.1 \$29.95
Screen Clean - Clears Up Video Distortion On Your Color Computer \$39.95
Set of Eight 64K RAM Chips w Mod Instructions \$49.95

DISK DRIVE PACKAGES, etc.

These Packages include the Radio Shack Disk Controller, Disk Drives with Power Supply and Cabinet, and Disk Drive Cable:

PKR #1 - 1 Single Sided, Double Density Sys. \$409.95
PKR #2 - 2 Single Sided, Double Density Sys. \$749.95
PKR #3 - 1 Double Sided, Double Density Sys. \$549.95
PKR #4 - 2 Double Sided, Double Density Sys. \$919.95
PKR #5 - 2 Queue Thinline Double Sided, Double Density Sys. \$499.95
PKR #6 - 2 Tandon Thinline Single Sided, Double Density Sys. \$599.95

PRINTERS

EPSON RX-80 \$375.00
EPSON MX-100 \$725.00
EPSON MX-80 \$395.00
EPSON PX-80 \$599.00

SERIAL BOARDS

MX Series \$119.95
RX PX-Series \$99.95
Spectrum MX-Series \$54.95

Mark Data Keyboards \$67.95

FLOPPY DISKETTES

	MEMOREX	VERBATIM
5" Soft Sector Disks		
Single Sided Single Density	\$2.40ea.	\$2.75ea
Single Sided Double Density	\$2.40ea	\$2.75ea
Double Sided Double Density		\$4.92ea
Plastic Storage Box		\$2.00ea
8" Soft Sector Disks		
Single Sided Single Density		\$3.75ea
Single Sided Double Density		\$4.10ea
Double Sided Double Density		\$4.75ea
Plastic Library Box		\$5.00ea

For Ordering Call TOLL FREE 1-800-338-6800

LAST OF PRODUCTION!!!

!! LIMITED QUANTITY !!

SWTPC 8212 & 8212(W)
Intelligent Terminals

New & Demo Models

At Discount Prices

Remaining Supply of
SWTPC 8212 CRT Terminals

EM-82 Video
Terminal

Emulates the 8200 Series from SWTPC



DATA-COMP

P.O. Box 794 HIXSON, TN 37343
1-615-842-4601

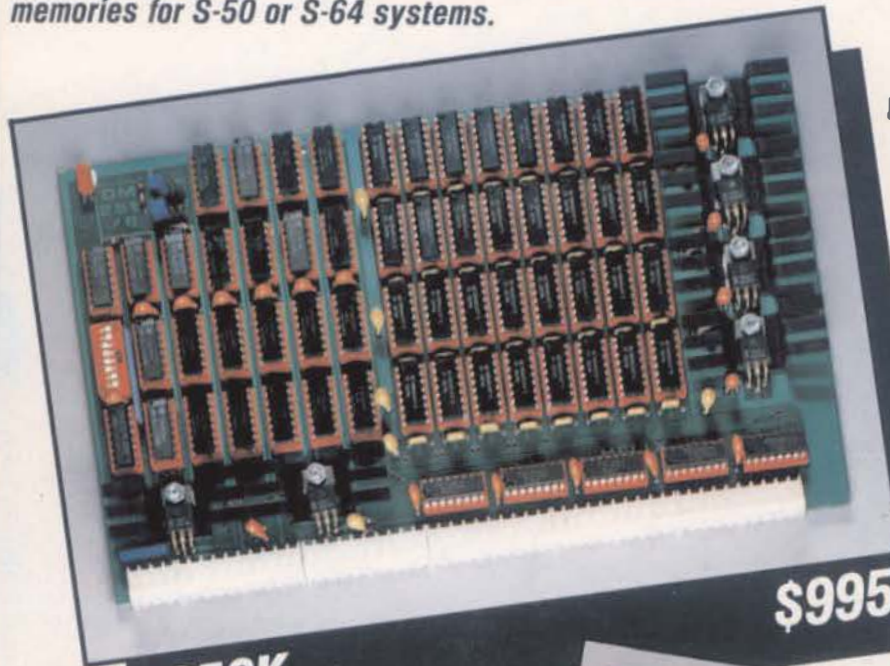
*FLEX is a trademark of Technical Systems Consultants
*OS9 is a trademark of Microware

Data-Comp--South East Media & 68 Micro Journal Are Divisions of CPI

KINGSTON SPRINGS TN 37082
P.O. BOX 87
MR. MICKEY FERGUSON
000422 A/E

MEMORY • MEMORY • MEMORY • MEM

Hazelwood Computer Systems presents a pair of advanced memories for S-50 or S-64 systems.



256K

\$995

Dynamic memory with guaranteed 2.5 MHz operation

The DM-256 utilizes the same proven refresh method as our original DM-64. This technique allows fully transparent refresh while maintaining full bus speed and eliminating difficult timing problems.

In addition, individual power regulators for each row of memory ICs enhance reliability by providing the ultimate in noise decoupling. Flexible addressing mode controls make the entire 256K available to OS9 Level 2 or UniFLEX. For FLEX, the furnished utility program HYPERDSK™ formats memory not used by FLEX as a virtual, superfast disk, thus tremendously improving total system performance.

With several hundred already in use, the DM-256 is the best choice for economical, reliable memory.

Assembled, tested, and burned in
Order: DM-256

Static memory with on-board battery backup

Utilizing the latest in low power CMOS static RAMs, the CM-64 has an on-board NiCad battery to maintain memory contents in the absence of bus power. Generous timing margins allow operation at bus speeds in excess of 2 MHz. Zener diode power loss sensing inhibits write operations to prevent data loss during power transitions. Write access can be explicitly controlled by means of an on-board jumper, a remote mounted switch, or an external logic signal.

This allows the memory to appear as ROM to the computer after the contents have been "programmed" using normal computer operations. The economical price makes the advantages of non-volatile memory available to any system.

Assembled, tested, and burned in
Order: CM-64



64K

\$395

HAZELWOOD COMPUTER SYSTEMS

907 East Terra, O'Fallon, MO 63366,

314-281-1055

HELIX™